Using Information Above the Word Level for Automatic Speech Recognition

Using Information Above the Word Level for Automatic Speech Recognition

Simon Alistair King

Thesis submitted for the degree of Doctor of Philosophy

University of Edinburgh

1998

Declaration

I have composed this thesis. The work in it is my own unless otherwise stated.

Simon A. King

Acknowledgements

I would like to thank my supervisor Steve Isard for being everything a supervisor should be. I received much support from my other colleagues at the Centre for Speech Technology Research. Paul Taylor and Helen Wright were directly involved in this work and I also had many helpful discussions with Alan Black and everyone else at CSTR – you know who you are.

Reuters very generously provided my funding without placing any restrictions on the work I did, for which I must thank Trevor Bartlett in particular.

And finally, I must thank my family and Fritha for all their support and encouragement over the years.

Abstract

This thesis introduces a general method for using information at the utterance level and across utterances for automatic speech recognition. The method involves classification of utterances into *types*. Using constraints at the utterance level via this classification method allows information sources to be exploited which cannot necessarily be used directly for word recognition. The classification power of three sources of information is investigated: *the language model in the speech recogniser*, *dialogue context* and *intonation*. The method is applied to a challenging task: the recognition of spontaneous dialogue speech. The results show success in automatic utterance type classification, and subsequent word error rate reduction over a baseline system, when all three information sources are probabilistically combined.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 The problem being addressed

- ***Speech recognition as a search problem***

Automatic speech recognition is a search problem. The search space is all possible sentences[1] in the language, and the solution we are looking for is the most likely sentence, given some observations and some constraints. Observations give us information about the utterance being recognised, and constraints express our prior knowledge. There are two distinct problems: defining the constraints so that the most likely sentence is the correct answer as often as possible, and ordering the search of the space so that it can be achieved in a reasonable amount of time.

The Viterbi algorithm (Forney, 1973) is an efficient method for solving the speech recognition problem formulated as a search through the space of all possible sentences. Because the search space is very large, we cannot afford to search very much of it for the solution. Typically, the search will proceed incrementally. A number of differing hypotheses about the solution will be considered in parallel. The number of possible hypotheses will typically be very large, so some *pruning* must take place: certain parts of the search space will not be explored. So, there are also two distinct possibilities for error: the constraints may not lead to the

[1]For the sake of simplicity, let us assume that all utterances are of sentences.

correct solution; and we may prune that part of the space which contains the correct solution.

This means that improving the constraints has two consequences: for a given amount of pruning, accuracy will increase; for a required accuracy, more pruning can be done. The constraints can be improved by better modelling of the observations available, or by finding new observations and modelling them.

- *Observations and constraints*

Generally, the only observation available is the speech signal, and the constraints take the form of stochastic generative models. The search problem is therefore to find the (sentence) model which is most likely to have generated the observations seen. In any recognition system with more than a few words vocabulary, the sentence models are in fact composed of smaller models. If these smaller models are of words, say, then a model of each possible sentence can be built by concatenating the appropriate word models. Practical systems typically use models of phone-sized units, and compose those word models using a pronunciation dictionary.

A system using *only* a model of the acoustic signal will typically not achieve good results. The solution it finds will not be very close to the correct one; word accuracy will be poor. More constraints are required to improve accuracy, and the obvious choice is a model of the language. This model constrains the way sentence models can be composed from word models, by assigning differing probabilities to different word sequences.

- *Restrictions of current approaches*

The acoustic observations are usually restricted to **spectral features**, which are typically vectors of cepstral coefficients. This parameterisation of the speech signal

describes the spectral shape. Typical analyses are based on a source-filter model, and it is assumed that the filter (vocal tract, lips, nasal passage) part of the model contains all the useful information and source (glottis) effects should be removed. Cepstral analysis captures this filter information, but effectively removes F_0.

Linguistic constraints are always used to improve recognition accuracy, but are generally restricted to models of *word* sequence probabilities – what we will refer to as a **language model.** These models generally take no account of context outside the current utterance.

- *More constraints*

It is widely accepted that speech recognisers could make more use of the information given by $\boldsymbol{F_0}$, but that using it in the same way as cepstral coefficients (that is, as just another component of the observation vector) does not work. This is because the spectral properties of the signal vary on a different (much shorter) timescale than F_0. In other words, the spectral properties of the signal vary at the segmental level and F_0 varies suprasegmentally – accents are placed on syllables, not phones. There are, of course, microprosodic effects which mean that F_0 does depend to some extent on the segment type; the use of this has been explored by Dumouchel and O'Shaughnessy (1993) and is discussed on page 98.

One area of current interest is the recognition of spontaneous dialogue speech – in the form of conversations between two people. Whenever utterances are part of a sequence, whether from a single speaker (a discourse) or a pair of speakers (a dialogue), there are clearly additional constraints of use in speech recognition. For example, we could model utterance sequences – with what we will call a **dialogue model**, or extend the language model to use context outside the current utterance. Recognition of dialogue speech has applications in human-computer interaction (HCI) and spoken language translation.

- *How to use the information*

Attempts to use these additional sources of information have often been somewhat piecemeal, often as add-ons to existing recognition systems. For example, F_0 has been used to guide syntactic parsing of speech recogniser output, but without the speech recogniser making use of this constraint itself. Another example might be the classification of utterances into types (statements and questions, for example) based on the word sequences from a speech recogniser, but without using utterance type constraints at the recognition stage. Utterance type classification is of particular interest for HCI applications because the type of an utterance carries information not necessarily present in the surface form (word sequence). If we classify utterances, we can condition language model probabilities on the utterance type.

- ***A novel method***

I propose that the classification of utterances into types is a way to incorporate constraints at the utterance level into speech recognition and that the automatic classification of utterances will lead to improved word accuracy. Further, I propose that to do this requires an *integrated* approach. Not only must we find the best models of the acoustic signal, language and dialogue, but we must combine these models in a probabilistic way that minimises pruning errors. For illustration, consider some *wrong* ways to go about this:

1. use a speech recogniser with a general purpose language model to find the most likely word sequence, then determine the type of the utterance based only on that word sequence using utterance type-specific language models.

2. use F_0 to determine the type of the utterance then do speech recognition using a language model for this type.

The first of these approaches is sub-optimal because the most likely word sequence is found using a sub-optimal language model; since the utterance is always of *one* of the types[2], using the appropriate language model *during recognition* should improve word accuracy. The second approach is worse; to make a *definite* decision about the type of an utterance based *only* on intonation observations is clearly not going to work, given the complexity of the relationship between intonational tune and utterance type. Every time the type is incorrect, an inappropriate language model will be used – leading to very poor recognition accuracy. As I will show, intonation has predictive power for utterance type, but works best in conjunction with other constraints.

In this thesis, I present a novel method which combines the information sources: **spectral features**, $\boldsymbol{F_0}$, **utterance type-specific language models** and a **dialogue model** in a probabilistic way. All of these components have parameters estimated from training data, and the acoustic signal is the only observation required[3]. A language model composed of utterance type-specific language models is employed, thus integrating utterance type constraints into the recogniser itself.

○ *An important assumption*

Because this thesis concentrates on a novel method for combining information sources at the utterance level, I will not address the problem of *segmenting* dialogues into utterances. The data used has been pre-segmented into utterances, and I will assume that, in a practical system, this could be done automatically.

[2]I will only consider utterance type classification schemes which can classify *all* utterances.

[3]The dialogue model also uses speaker identity, but this is assumed to be easily derived from the acoustic signal, since the recordings use a separate channel for each speaker.

- *Outline of the method*

The method can be summarised thus:

- Utterances are categorised into a number of dialogue-theory motivated types.
- Unknown utterances are classified as one of these types using
  - F_0 information, via a probabilistic model of intonation,
  - language models which account for utterance type,
  - dialogue structure constraints.
- All these information sources are combined in a probabilistic framework, which avoids hard decisions and allows weighting of the relative importance of each source.

The strategy is therefore to determine the *type* of each utterance and then determine the most likely word sequence, *given* that *type*. If language models are to be used for predicting the utterance type, this will involve doing speech recognition, so the task of determining the most likely word sequence will simply become that of picking the recogniser output which was produced using the language model of the appropriate type.

- *Originality of the method*

The method is original because these information sources are combined at the lowest level possible – the utterance level – and word accuracy is *directly* improved, without the need for further processing (syntactic analysis, for example).

One goal of the work is of course to improve word recognition accuracy for speech recognition of spontaneous dialogues, but another goal is the actual classification of utterances into types. I will show that these two goals can be achieved

simultaneously: by finding the most likely utterance type sequence for a dialogue, word accuracy is improved over a baseline system which does not classify utterances.

Although utterance type categorisation is not a new idea, simultaneous utterance type classification and word sequence recognition is. The method is general in that it does not rely on the particular categorisation system used or the method(s) of utterance type classification. The probabilistic combination of information sources used for type classification allows assessment of the contribution made by each source; results are given using various combinations of the information sources listed above.

To test the hypothesis that utterance type classification is a useful way to combine information sources, an experimental speech recognition system was built. This system consists of a number of components, reflecting the number of information sources used.

1.1.1 About this thesis

- ***Structure of this thesis***

The structure of this thesis reflects the process of building the experimental system. Each component of the system was designed and evaluated, followed by integration and evaluation of the whole system.

Utterance type classification will be considered first, as it is the foundation for all other work. Chapter 2 describes the theory of utterance type classification, surveys the literature, and introduces the classification system of Carletta *et al.* (1995).

The following chapters each examine a component of the system and give the design choices which must be made; the choice made in each case is based on the literature and the results of the experiments which are described in this thesis.

A baseline speech recognition system is established in chapter 3. The state of the art in speech recognition is surveyed; the results of this survey are used in the design of the baseline system, which is then evaluated experimentally and shown to perform at least as well as other current systems on comparable data. The baseline system not only provides a reference point (in terms of word accuracy) for the new method, it also provides the acoustic models which will be used later.

Chapter 4 concerns language modelling. The state of the art is reviewed, and the building of language models for the experimental system is described. The language models are assessed using data, and the best performing model is selected for use in the system. In chapter 5, intonation is examined. Several frameworks for describing intonation are considered, and their usefulness for the automatic recognition of intonation is evaluated. One framework is chosen, and evaluated experimentally. Dialogue is covered in chapter 6, and the problems of modelling it are described. Here, a database of recorded dialogues is introduced which will be the data for the experimental system. Experiments relating to the *design* of components are documented in the relevant chapter, and experimental evaluation of the integrated system is in chapter 7.

The whole system is then used for utterance type classification and speech recognition; this is covered in chapter 7 . The results show that the method of combining information sources is successful, and gives improvements in accuracy over the baseline system. Conclusions are drawn in chapter 8. The implications of the results using the experimental system are considered, with an analysis of where improvements could be made.

- ***Distribution of the work***

The system described in this thesis is the work of several people. Those parts which are not the work of the author are:

- intonational event detector and labeller (Paul Taylor)
- utterance type intonation models (Helen Wright)

A sufficient description of these components is contained in this thesis, and reference is made to published descriptions. Stephen Isard also contributed to all parts of the system, and valuable contributions were made by Hiroshi Shimodaira. Everything else, namely the speech recogniser, language models, dialogue model and system integration is the work of the author.

The EPSRC[4] funded project "Intonation and dialogue models as constraints in speech recognition" (CSTR, October 1993 - March 1997) – known as ID4S – set out to improve speech recognition results by using the interaction between intonation and dialogue context as a constraint. At the time the project was conceived, the intention was to use an existing CSTR speech recogniser, but shortly after it started a decision was taken to switch to the widely used HTK toolkit, in part to show that the contribution, if any, of the novel components would be relevant to mainstream speech recognition work of the day. At this point I started to build the experimental system described in this thesis. The ID4S project proposal left the method by which intonation and dialogue models would be used as constraints in speech recognition as an open question. This thesis contains an answer to that question.

[4]Engineering and Physical Science Research Council

- ***Published work***

Some of the work contained in this thesis has been published. I am co-author of the following publications: (Isard *et al.*, 1995; Taylor *et al.*, 1996; Taylor *et al.*, 1997b; Taylor *et al.*, 1998,pending). The intonation component is described in (Wright & Taylor, 1997)

1.2 Background: spoken language systems

One of the goals of the work described here is to improve speech recognition accuracy, the simplest measure of which is word accuracy. To do this, a novel approach is used in conjunction with a conventional speech recogniser. To show the contribution made by the novel technique, we establish a *baseline* system which uses "state-of-the-art" technology to ensure that the improvements obtained are genuine; improvements to poor systems mean little.

Throughout this thesis, it should be assumed that the speech recogniser is a component of a spoken dialogue system, such as Verbmobil (Wahlster, 1993) or Trains (Allen *et al.*, 1995; Ferguson *et al.*, 1996). Therefore, the first thing to do is to review the state of the art in speech recognition, and in the other components of spoken dialogue systems.

1.2.1 Tasks which need speech recognition

What are we doing speech recognition for? We can categorise speech recognition tasks into passive, active and interactive. Dictation is a passive task, command and control is an active one and interactive tasks involve two-way communication.

- ***Dictation***

Speech recognition systems for dictation tasks have been around for a number of years and have evolved from early systems requiring what amounted to isolated word input, to those which allow "naturally" spoken input. The possibilities for improving these kinds of systems include things like noise robustness, speaker adaptation, language model and domain adaptation and so on. As these systems are required to recognise more spontaneous speech, the novel method introduced in this thesis will find applications. Dictation is of course, not a dialogue, but a discourse. There is still the possibility of modelling effects across utterances using a discourse model.

- ***Command and control***

Using speech recognition for command and control of systems requires severe limitations to ensure reliable operation. If commands are accepted without question, the speech recogniser must be either very accurate, or able to reject difficult utterances. Adding some form of clarification dialogue could alleviate this problem - the task then takes the form of a goal oriented dialogue.

- ***Goal oriented dialogues***

For more complicated tasks than dictation, and for command/control tasks where some interaction is required, a *goal oriented dialogue* can be the most effective form of communication. Typical goal oriented dialogues are *co-operative* – the participants are trying to achieve the same goal, which may be completion of some task, or transfer of information.

This sort of task is not limited to human - computer interaction. In other situations, such as the Verbmobil translation system (Wahlster, 1993), the conversation may be between two people, with the machine following the dialogue.

1.2.2 Dialogue systems

Here I review the state of the art in spoken dialogue systems by considering some well known systems. Such systems are generally large and complicated, and often quite task- or domain-specific. I will concentrate on the speech recognition aspect of the following systems, and how dialogue context and/or intonation are used to improve recognition accuracy.

One of the most obvious problems in comparing these systems, is that they all use their own data, which is invariably collected specially. The databases used can be compared in terms of vocabulary size, speech quality, whether the speech is spontaneous or read text and on the "difficulty" of the task. I will give this information for the systems I consider, to allow comparison with the experimental system used to investigate the new method introduced in this thesis.

- ***Trains***

Figure 1.1, taken from (Allen *et al.*, 1996), shows the TRAINS (Allen *et al.*, 1995; Ferguson *et al.*, 1996) scenario. The task is route planning, where the user must interact with the computer to select the best route for trains travelling between cities. The TRAINS system is composed of "off-the-shelf" components (speech recogniser and synthesizer) and specially designed ones, such as the parser, discourse manager and

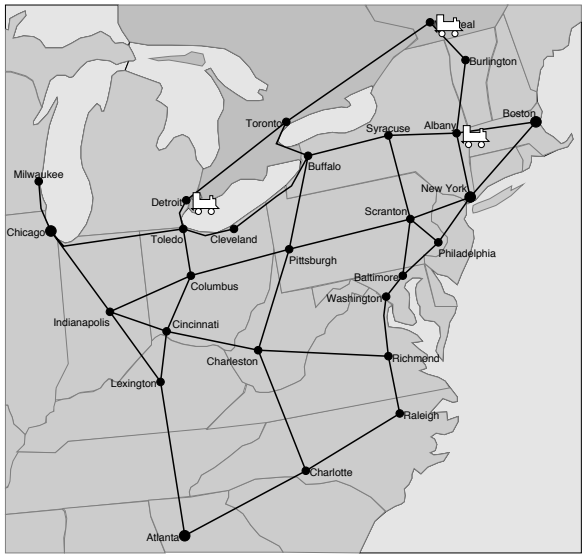


Figure 1.1: The TRAINS scenario

reasoning components. Interaction with the system is multimodal: the user can type in or speak their requests, and the system responds through speech and a visual display screen. Only the speech recognition and dialogue components are of interest here.

The vocabulary size in TRAINS is around 1000 words, which is very close to the vocabulary size for the DCIEM dialogues which will be used in this thesis (see section 6.3). The speech was collected using high-quality close-talking microphones and recorded on DAT – therefore it is of a very high quality.

- *Speech recognition*

The speech recogniser in the TRAINS system is the Sphinx-II system from CMU (Huang *et al.*, 1993). The system achieves word error rates of 30% and 20% using language models trained on ATIS and TRAINS data respectively.

- *Post-processing*

The speech recogniser output is post-processed to reduce errors. This approach has the advantage of allowing the post-processor to use methods not possible or practical during speech recognition, such as higher order N-gram language models. The post-processor is trained using pairs of correct transcriptions and corresponding speech recogniser output, it can therefore learn the kinds of errors the recogniser makes. In other words, the post-processor models the speech recogniser as a noisy channel, and attempts to correct the errors it introduces.

- *Speech acts*

Identifying "intent" is used for disambiguation, both for spoken and other communications modes (using a mouse, for example). The dialogue model uses *speech acts* which are determined by a combined syntactic and semantic parser. Some

very simple cues to speech act are used, such as common word sequences ("I want to...", for example). The claim is that this approach is robust to speech recogniser errors, but clearly it is also very limited, since these short phrases are by no means the only cue to the speakers' intent.

- *Dialogue manager*

Discourse state is tracked using a stack. The stack supports just two operations: push and pop. Pushes open new dialogue segments, and pops complete them. A simple example would be the effect of the statement "No" whose speech act is "reject" which causes all information in the preceding dialogue segment to be removed from the stack. Nesting of segments results in consecutive pushes. This is related to the theory of conversational games, described on page 112, where games are dialogue segments which open and close, and can be nested.

The dialogue parser (Core & Schubert, 1996) in the TRAINS systems accounts for spontaneous speech and dialogue effects using additional speech segment types: "lulls", "interpolations" and "backtracking". Lulls are the most interesting of these, and are used for pauses, back-channel speech and change of turn boundaries. No use is made of prosody, but the notion of lulls is intended to allow for some account of prosody to be included eventually.

- ***Verbmobil***

Verbmobil (Wahlster, 1993) is a speech-to-speech *translation system* for face-to-face dialogues between two people – see figure 1.2. A certain amount of human-machine interaction is required for clarification and other purposes. The demonstration task is that of arranging business meetings. The system is complex and includes speech recognition (Plannerer *et al.*, 1994; Reichl & Ruske, 1995), NLP, machine translation, generation and speech synthesis (King *et al.*, 1997) components. This is one of the few systems to have explored the use of prosody extensively for a variety of purposes (Hess *et al.*, 1996; Strom, 1995; Strom *et al.*, 1997). From the acoustic signal, accents, boundaries, focus and sentence modality are estimated, and used at various stages, from the morphological parser up to the semantic parser.

Figure 1.2: The Verbmobil scenario

The vocabulary size for the demonstrator system for Verbmobil is 1300, which, like the TRAINS vocabulary, is very similar to the DCIEM corpus which will be used in this thesis (see section 6.3).

- *Accents and boundaries*

A system of four types of boundary and three types of accent is used, which is similar to ToBI[5] (Silverman *et al.*, 1992). With various constraints, only six dif-

[5]Tones and Break Indices - a system of numerical scales for representing prosodic elements.

ferent combinations of accent and boundary are allowed, and these are recognised using a neural network. The input features to this network are syllable based (a segmentation is available from a speech recogniser), and comprise over 240 parameters of the current and neighbouring syllables including F_0, energy and lexical accent. An average classification rate for four classes of boundaries of around 80% is reported for spontaneous speech, given syllable nuclei locations.

- *Focus*

Focus, defined in Verbmobil as *prosodic* focus, which means the word with the most prominent accent, is useful information for semantic and higher level processing. Elsner (1997) presents a rule based method for detecting focus from F_0. Sentence modality and phrase boundaries are also available cues. This work appears to be less well developed than the accent and boundary classification, and only preliminary results are reported in (Elsner, 1997).

- *Sentence modality and use of dialogue context*

Verbmobil follows the dialogue between two participants, in order to build a model of dialogue context which is required to aid translation requests. This is done in terms of *Dialogue Acts* (DA). Therefore, the speech must be segmented into DA units, and each one classified as one of the DA categories (DACs): greeting, confirm date, suggest place and so on. There are 18 categories altogether.

The speech is assumed to be already divided into *turns*. Warnke *et al.* (1997) describe an approach to segmentation and classification of DA units. Available as input for this task is a word lattice generated by a speech recogniser. A scheme similar to the accent and boundary detector described above is employed to detect DA boundaries. Their first approach is two-step: segmentation followed by classification. The segmentation is based on prosodic features, as used in the ac-

cent/boundary detector, and N-gram language models. These language models estimate probability of dialogue act boundaries from word sequences. Subsequent classification uses DAC-specific language models (which are also N-gram models) and a DAC sequence model (again, an N-gram model) which models DAC sequences only *within* turns.

A second, integrated, approach performs segmentation and classification in a single step. The probabilities from prosody, the DA boundary language model, DAC-specific language models, and the DAC sequence model are combined as a weighted sum in the log domain, and an optimal combined segmentation/classification is found in a single search step.

The accuracies reported were similar for both the two step and integrated approaches, at around 53%. Interestingly, in the integrated approach, the DAC sequence model (that is, dialogue model), can be omitted with little loss in accuracy. This may be because this model only accounts for DAC sequences within the current turn, and there are typically only a few DA units per turn. Although the Verbmobil system uses context *across* turns for semantic processing, these constraints are not applied at the speech recognition level. The result of this is the small contribution made by the dialogue model to DAC classification accuracy.

- *Eliciting easy to recognise responses*

As we will see later (on page 106), the dialogue can be manipulated by one or both participants. This is most commonly done by asking "leading" questions – in which the intention is to elicit an easy-to-recognise response. For example, if we were fairly sure that the other participant was going to answer "Yes" to some question, we could phrase that question in a way which prompts intonational marking should the answer be "No".

question	You **are** free on Monday, aren't you ?
answer	Yes. *(not intonationally marked)*
	No, I'm **not!** *(intonationally marked)*

The two possible answers to the questions are now not only distinguished by word sequence, but also by intonation, making them easy to differentiate.

- *Conclusions*

Verbmobil has shown that prosody "favourably contributes to the overall performance of speech recognition" although "Even if the incorporation of a prosodic module does not significantly increase word accuracy, it ... reduces overall complexity" (Hess *et al.*, 1996).

- ***Other systems***

- *Gemini*

Gemini (Dowding *et al.*, 1993) is a natural language system designed for spoken language. The Gemini system uses a variety of grammars using the formalism of the Core Language Engine (CLE) (Alshawi, 1992). Dowding *et al.* (1993) makes no mention of the speech recogniser, or what grammar it uses. The aspects of Gemini which are of interest here are two rule-based modules: the first is used for "glueing" together fragments of sentences and the other is used for repair. This approach, of post processing the output of a speech recogniser, is common in spoken language systems, but the degree to which higher level constraints are applied *at* the speech recognition level is crucial. Typical syntactic and semantic parsing is done with language models not suitable for speech recognition, so such constraints are difficult to apply. Dowding *et al.* (1993) suggest that the syntactic and semantic parsing be "interleaved" with the recognition – that is, the parser should be able to process sentence fragments as recognition proceeds, and pass

constraints back to the speech recogniser.

- *Spoken Language Translator*

The Spoken Language Translator (SLT) (Rayner & al, 1993) from SRI is a prototype system for translating spoken language from a restricted domain. The prototype translates ATIS-domain utterances from English to Swedish. The system is particularly interesting because it was constructed from existing pieces of speech and language software, rather than custom built as was the case, for example, in Verbmobil (Wahlster, 1993). Such approaches can be suboptimal because the interfaces between components are frequently weak points. For example, using a speech recogniser to give a single sentence as output will not be as good as outputting a word hypothesis lattice or N-best list. The SLT passes N-best lists out of the speech recogniser. Higher level processing uses the unification grammar formalism of the Core Language Engine (CLE) (Alshawi, 1992), which is not suitable for direct use in a speech recogniser, so a post-processing approach must be taken.

Chapter 2

Utterance type classification

2.1 Introduction

The novel, integrated approach to dialogue speech recognition introduced here combines information from spectral parameters, F_0, language model and dialogue context in a probabilistic way. In chapter 1 I described speech recognition as a search problem, with the goal of finding the most likely word sequence for an unknown utterance, given a set of observations and constraints. Within an utterance, this means optimising over all possible word sequences. The information provided by F_0[1] and dialogue context is not at the word level, but at the utterance level. That is, F_0 and dialogue context are used to estimate probabilistically the *type* of an unknown utterance. Therefore, the optimisation is not over the *word* sequence for a *dialogue*, but over the *utterance type* sequence. That is, we will find the most likely *utterance type* for each utterance in a dialogue. The hypothesis is that this will lead to improved word accuracy.

[1] I will use F_0 and intonation interchangeably, since F_0 will be stylised as a series of pitch accents, boundaries and connections – see chapter 5

- *Organisation of this chapter*

This chapter starts with the theory of utterance type classification, and introduces the scheme which was adopted. There was, in fact, little choice of utterance classification schemes for several reasons: there are not many schemes to choose from; the database chosen was already coded using a particular scheme; recoding the database with a new scheme would be far too costly. It must be pointed out that, despite this, the method does not rely on this particular classification scheme. I show how constraining utterances to be one of a fixed set of types allows the imposition of further constraints such as dialogue context and speaker rôle. Since there was effectively no choice of classification scheme, I then attempt to improve the scheme by small modifications, namely merging and splitting some of the types.

2.2 Theory

Now that I have established utterance type classification as the goal, a set of utterance types must be chosen. I have postulated that this classification will both improve speech recognition word accuracy, and be an end in itself, as in (Garner & Hemsworth, 1997) for example. For the type set to be useful, the types must relate to the *content* **and** the *dialogue rôle* of utterances. The automatic classification of utterances is intended to be useful in spoken dialogue systems, so the choice of a type set which reflects the role of the utterance in a dialogue seems a good choice. The theory of conversational games (Power, 1979) will be introduced on page 112, and this theory defines a type set which meets these criteria. It was already thought that the move types would be intonationally distinguishable.

instruct	a direct or indirect request or instruction *e.g. Go round, ehm horizontally underneath diamond mine ...*
explain	provides information that the speaker believes to be unknown by the other participant *e.g. I don't have a ravine.*
align	checks that the other participants understanding aligns with that of the speaker *e.g. Okay?*
check	asks a question, to which the speaker believes the listener knows the answer *e.g. So going down to Indian Country?*
query-yn	a yes–no question *e.g. Have you got the graveyard written down?*
query-w	a question containing a wh-word *e.g. In where?*

Table 2.1: Game initiating move types

2.2.1 Utterance types

There is no predefined set of utterance types, unlike the problem of word recognition. The choice is crucial, as different categorisation systems vary in their ease of recognition (from intonation or word string, for example) and their usefulness in speech recognition.

The starting point here is the set of 12 move types (Carletta *et al.*, 1995) for the Map Task. *Moves* come from Conversational Game Theory, which is described on page 112; all we need to know here is that moves typically contain single utterances, and that move type describes the utterance's rôle in the dialogue. Moves fall in to two types: those which can initiate conversational games, and those which cannot. The move types and their rôles are described[2] with examples

[2]Descriptions and examples are adapted from Taylor *et al.* (1998,pending)

acknowledge	acknowledges hearing or understanding *e.g. Okay.*
clarify	clarifies or rephrases *old* information *e.g.* { ***Other participant said:*** *So you want to go ... actually diagonally so you're underneath the great rock.*} *Diagonally down to uh horizontally underneath the great rock.*
reply-y	responds to a query-yn, check, or align; usually indicates agreement *e.g. Okay.* *e.g. I do.*
reply-n	responds to a query-yn, check, or align; usually indicates disagreement *e.g. No, I don't.*
reply-w	elicited response that is not to clarify, reply-y or reply-n; can provide new information that is not easily categorised as positive or negative *e.g.* { ***Other participant said:*** *And across to?* } *The pyramid.*
ready	indicates completion of previous game and that a new game is about to begin *e.g. Okay.* *e.g. Right.*

Table 2.2: Other move types

from the DCIEM[3] version of the corpus (Bard *et al.*, 1995) in tables 2.1 and 2.2; their frequency in that corpus is given in table 2.3 (taken from table 4.3 on page 76).

The Map Task corpus consists of spontaneous, goal-oriented dialogues between two participants. Both participants have copies of a map, one of which has a route plotted on it. The participant with this map is called the "instruction giver" and must describe the route to the other participant – the "instruction follower". To

[3]The Defence and Civil Institute of Environmental Medicine (Canada) sponsored version of the Map Task corpus using Canadian speakers of English.

make the resulting dialogues more interesting, there are slight differences in the landmark features between the two maps. Chapter 6 contains a full description of the Map Task corpus.

The distribution of move types, shown by table 2.3, is far from uniform. However, the distribution is not so uneven as to result in a very low information content (cf. the ToBI system for intonation labelling described on page 84). For the 12 move types, the task perplexity is 9.1 – this is the perplexity of the unigram language model (see page 119). A perplexity of 12 would indicate that all move types were equally likely. Perplexity is defined on page 49.

move type	*frequency*	*move type*	*frequency*
instruct	1407	acknowledge	2607
explain	733	clarify	246
align	319	reply-y	1020
check	598	reply-n	262
query-yn	703	reply-w	331
query-w	262	ready	784

Table 2.3: The 12 move types used in the Map Task dialogue coding and their frequency in the DCIEM corpus (training set)

2.2.2 Other work on dialogue move classification

As I have noted, the identification of the move type of utterances in a dialogue is an end in itself, and not simply a way to improve word accuracy. This task is related to topic identification, in that the word sequence can be used to identify some very broad semantic property of the utterance.

Bird *et al.* (1995) report results for move type identification for the Map Task (the original version (Anderson *et al.*, 1991)). Techniques for topic spotting are adapted for move type recognition; these are all based on utterance fragments – short word phrases found automatically from training data. For example, the fragment "and then" is relatively common in *instruct* moves. The results reported

are based on the **correct** utterance transcriptions (rather than automatically recognised ones, as used in the work in this thesis). Between 36% and 38% of moves are correctly recognised, depending on the exact method used. They report a similar pattern of classification accuracy across move types as we do (refer to table 7.2 on page 144) – with *acknowledge* moves being much easier to recognise (typically 70% accuracy) than, say, *explain* (30% accuracy). The corresponding figures from table 7.2 are 80% and 37% using *automatic* speech recognition (and the DCIEM version of the corpus).

Garner and Hemsworth's work (1997) follows from (Bird *et al.*, 1995). The move type[4] classification rate reported is now around 52% for the Map Task corpus (HCRC version). Results are also given for report topic identification using the LOB corpus, where around 55% of report topics were correctly identified (number of topic types not reported !).

2.3 Additional constraints

Having described speech recognition as a problem of search under constraints, and having decided to use these constraints via utterance type classification rather than directly at the word recognition level, there is no limit on what these constraints might be. So far I have only considered modelling the sequence of move types (with a dialogue model), but more information is available.

2.3.1 Further linguistic constraints

The move type-specific language models take account of the different syntax of each move type. In the Map Task dialogues, there are a set of maps, each with different features. This means that the language models could be made "map type" specific. There would clearly be a problem with lack of data if this was

[4]As ever, there are 12 move types.

done explicitly. However, only one class of lexical items changes – geographical features – and these items can be determined *a priori* from the map. So it would be possible to use this information to improve the language modelling further. One way might be to use class-based language models, with a *feature* class which can be filled by whatever entities are present on the current map. We could refine this and divide entities into more specific sub-categories, such as natural (lakes and forests, for example) and man-made (cottages, for example).

2.3.2 Non-linguistic information

By "non-linguistic" I mean sources of information not present in either the acoustic signal or surface form of utterances. Examples are the speaker's rôle in the dialogue and non-verbal communication.

- ***Speaker identity***

By "speaker identity" I mean specifically the rôle of the speaker in the dialogue and not their gender, age or other characteristics. Typically, each speaker will have his or her own microphone (this is always true for telephone conversations!) and therefore separate acoustic signals for the two speakers are available. From this, I will assume speaker identity is easily and accurately determinable with 100% accuracy.

- ***Visual cues***

In face-to-face dialogue, further visual cues are available. Simple cues such as eye contact and head movements add significant information. It can be seen from the Map Task corpus that when participants have eye-contact, dialogues are completed in significantly fewer moves. The HCRC Map Task dialogues do in fact have eye movement coding, but the DCIEM dialogues do not. This coding is very simple, and was done via video recordings of the dialogues. The coding indicates,

for each participant, whether they are visible on camera, and, if so, whether they are looking down at the map or up at the other participant (or at a dividing screen for the no eye contact condition). In auditory channel only dialogues, visual cues are replaced by *back channel* communication (ums and ahs, for example). Some backchannel sounds were used in this work, and they were treated as lexical items. Further use could be made of things like pause duration[5].

2.4 Regrouping the move types

It would seem unlikely that the 12 move types (what I will call the *original set of move types*) defined by Kowtko *et al.* (1993) are the best possible ones for our purposes. From experimental evidence, given in section 7.3.1, the *original set of move types* does work reasonably well, so attempts to find a better set of move types started from the original set. Experimental results can be found in section 7.3.4.

2.4.1 Constraints

Some simple utterance properties such as overall duration may provide a simple (easy to recognize) and effective (reduced perplexity language models) way to subdivide utterances into types. However, we decided that only utterance properties with a definite *rôle in the dialogue* would be used; this is motivated by the goal of simultaneous word recognition and utterance type classification, and the desire to classify utterances in a way useful for higher level processing (see section 2.2).

The inability of intonation to distinguish some pairs of move types lead to some merging of types. Evidence in (Hockey *et al.*, 1997) for the context-sensitive nature of some of the original move types lead to splitting some of the types as detailed below.

[5]Rather than mere pause *presence*.

2.4.2 Merging

Whilst merging any of the *original* move types clearly reduces the information content (entropy) in dialogue modelling terms, there are strong reasons for doing this on intonational and language modelling grounds. If a pair of moves cannot be distinguished intonationally, then the intonation recogniser is not going to add useful information to distinguish them – that is, it will not give an entropy reduction. Likewise, if two move types have very similar language models, then there is no advantage in distinguishing them, and we will be needlessly decimating the training data. But, as have already mentioned, the original move type set offers a useful degree of dialogue description; any type merging will reduce the usefulness of utterance type classification for subsequent processing.

- ***Intonation***

Some move types are apparently indistinguishable from intonational evidence – that is, using our automatic utterance type classifier with only intonation as input (see section 7.3.1). Whilst there are strong intonational contrasts between clarifications and questions, there do not appear to be any between clarifications and explanations. This alone is not enough to justify complete merging of explain and clarify. We could decide not to distinguish between them intonationally but to still use distinct language models. In speech recognition terms, we might describe this as *tying* the intonational models for explain and clarify.

- ***Language modelling***

If our language models make no use of whether information is given or new, then there will be little difference between the language models for explain and clarify, since clarifications are basically explanations using old information (such as already mentioned entities); there will possibly be more pronouns in clarify

utterances. Merging these two language models makes sense.

2.4.3 Splitting

The motivation for splitting some move type categories is that the language models, and possibly intonational patterns, for certain types are context-dependent. For example, the grammar of replies will depend not only on the type of reply, but on the type of move they are replying to. We assume context-dependence is from the left only, that is the grammar of a move type does not depend on the *following move*. The right-context effect, of prompting a certain type of response, is already sufficiently described in the move type set: for example, the two types of questions (yes/no and wh).

2.4.4 An alternative move type set

Based on the above observations and intuitions, we performed a combination of move type merging and splitting as follows:

- explain, clarify and instruct were merged
- reply-n was split into three types according to the type of the preceding move: query-yn; align or check; other
- reply-y was split into three types as per reply-n
- acknowledge was merged with the type "reply-y preceded by align or check"
- align, check, query-w, query-yn, ready and reply-w were unchanged

The result was a set of 13 move types. Experimental results for this move type set can be found in section 7.3.4

Chapter 3

Speech recognition

3.1 Introduction

This chapter covers the design of the baseline system. This system serves two purposes: to provide a reference for the system using utterance type classification, and to provide the acoustic models (Hidden Markov Models) for that system. The novel method introduced in this thesis does not require modifications to these models; therefore they can be taken directly from the baseline system, which means that the models can be trained on all available training data, regardless of utterance type.

- *Organisation of this chapter*

A review of the state of the art in automatic recognition of speech is given, with an emphasis on spontaneous, dialogue speech. This leads to the design of the baseline system, which is then built and tested. Its performance is shown to be at least as good as other current systems under comparable conditions (such comparisons are difficult because it is hard to quantify the differences between the domains and data used by each system).

3.2 The state of the art

3.2.1 Systems

Speech recognition systems can be classified in several ways. They could be put in to one of two classes: large vocabulary, domain-specific systems or small to medium vocabulary robust systems. The problems facing each of these types of systems are quite different. Alternatively, we can divide systems into those which recognise "read text" speech and those which recognise genuinely spontaneous speech. These two divisions coincide, since large vocabulary systems, such as that in (Woodland & Young, 1993), generally only recognise what amounts to "read text" speech which is noise-free, fluent and clearly spoken. The reasons for this are the difficulty of recognising large vocabularies, the problem of collecting enough data to train acoustic models, and, more especially, language models. For read text, such as newspaper articles, very large amounts of training data for language modelling are easily obtained.

On the other hand, spontaneous speech recognition systems often operate with only a limited vocabulary. One of the most challenging tasks at present is the Switchboard corpus of spontaneous telephone conversations (Linguistic Data Consortium, 1993-7). Although not explicitly constrained, the vocabulary is of only moderate size (around 25 000 words[1]) because the conversations are about everyday subjects.

- ***Large vocabulary systems***

The major problem facing systems with large or very large vocabularies is the sheer size of the search space. Language models for vocabularies of tens of thousands of words have potentially very large numbers of parameters and can have perplexities (defined on page 49) in the hundreds. Estimating the parameters of these language

[1]Counted in phase 1 transcripts.

models becomes difficult due to the amount of data needed even for bigram or trigram models. Many of the techniques used to overcome these problems are applied at recognition time, and include:

- Two pass approaches: a simple language model is used to produce candidate word lattices (in other words, to narrow down the search space) which are rescored using a more sophisticated language model
- Advanced search techniques: exploring the search space in a more intelligent way, and representing hypotheses efficiently, as in stack decoding and time-asynchronous search, for example (Jelinek, 1969; Bahl & al, 1988; Paul, 1992)
- Hybrid systems: separating phone probability estimation and search, as in (Renals & Morgan, 1992) and (Robinson, 1993) .

- ***Spontaneous speech systems***

The problems facing systems which recognise spontaneous speech are slightly different. The size of the vocabulary does not cause problems in terms of language model complexity, but the lack of training data does make acoustic and language models hard to estimate. Techniques for overcoming difficulties due to lack of training data can be applied when building the system (estimating language models, for example) rather than when recognising speech.

Language model estimation requires a variety of techniques to make the most of limited training data. Additional constraints may be required to increase robustness, and special treatment of spontaneous speech phenomena such as disfluency, ungrammaticality, extreme co-articulation and variable pronunciation is needed.

One approach to the problem of recognising spontaneous speech has been wordspotting. By only recognising words of interest to the system, a degree of

robustness to spontaneous speech effects is achieved, at the expense of limiting the recogniser to signal the presence of words rather than whole utterances. In systems like Gemini (Dowding *et al.*, 1993), a form of wordspotting is used in the semantic interpreter, which uses the presence of key words or short phrases to determine speech acts. In a very limited domain, such an approach may work, but, in general, speech act is not uniquely signalled in the surface form of an utterance, and methods such as that described here which combine estimates of utterance type from multiple sources are more likely to work and be robust to variations in surface form. The new approach described here places no restriction on the estimators of utterance type.

- *Dialogue speech*

In conversations between two (or more) people, or indeed human and machine, there are some properties which are absent in read text. The prosody and intonation are likely to be more "interesting" and informative, there are higher level constraints such as utterance purpose (illocutionary force), turn taking, and so on. Therefore investigation of the use of these additional sources of information will be easier with spontaneous dialogue speech since the effects are likely to be stronger than in, say, read text. In other words, the contribution of intonation (for example) to speech recognition is more easily assessed using data certain to contain informative intonation. The fact that intonation carries useful information has been shown by Kowtko (1996) using the Map Task corpus (see page 93).

3.2.2 Benchmark tasks

In order to evaluate performance, some standard task is required – a *benchmark*. Since much research in recent years has been on large vocabulary (RM, WSJ, Switchboard) or speech in noise tasks (Noisex, for example), these types of databases are readily available. Unfortunately, for the developer of spoken di-

alogue systems, the choice of corpora is rather limited. However, it is better to choose an existing, publicly available database than to generate one – because this is time and resource consuming and makes comparison with other work difficult.

- ***Large vocabulary recognition***

With the recent emphasis on large vocabulary recognition (LVR), several large, and some *very* large, corpora exist to support this kind of work. I will briefly examine them here, and show why, although they typically contain very substantial amounts of data for training models, they are not suitable for our purposes.

  - *Resource Management*

This is a one-way command and control task – there is no interaction with the computer. Since the data is actually **read text** from a fixed set of sentences, the intonation is very uninteresting and not especially informative. The data does not contain any disfluencies or non-speech sounds.

  - *Wall Street Journal*

This is another read text task, but with a much larger vocabulary than the Resource Management task. Again, read text has uninteresting intonation which adds little information. However, this task does have the attractive property of very large amounts of textual training data which allows the estimation of more complex language models.

- ***Spontaneous dialogue speech corpora***

Spontaneous speech is cheap to collect: for example, the Switchboard corpus is simply recordings of telephone conversations. However, it needs to be transcribed by hand, which is expensive, especially if detailed labelling of disfluencies (see

section 4.5.1) is required. Furthermore, there are inevitably errors in transcription and places where the speech is ambiguous (and therefore transcription is difficult), or where arbitrary decisions must be made.

- *Switchboard*

The Switchboard corpus consists of spontaneous speech gathered over the telephone network, loosely restricted in domain but not vocabulary. The conversations are social calls, without any ultimate goal. Speech recognition work on this corpus is still ongoing (CLSP, 1997) and the combination of telephone-quality speech, lack of goal in the dialogues[2] and large inter-speaker variation make this corpus a little *too* challenging for development work! However, the technique described in this thesis *has* subsequently been applied to the Switchboard corpus by Taylor *et al.* (Jurafsky *et al.*, 1997; Shriberg *et al.*, 1998).

- *Map Task*

This corpus was collected specifically for dialogue-related research, and as such, great care has been taken to effectively limit the vocabulary size through the design of the task. The dialogues have a specific goal, so they follow much more constrained patterns than those from the Switchboard corpus, and therefore lend themselves more to dialogue modelling. Other variable factors have been largely controlled: the speech signal is of a high quality with no background noise; speakers have reasonably similar accents; the labelling is of a high quality. Although somewhat artificial, this task was therefore thought to be a more practicable proposition. A full description of the Map Task can be found in chapter 6.3. A significant amount of other work has been done using the Map Task corpus – in both its original and DCIEM versions. In particular, Power's theory of conver-

[2]Which leads to a lack of global structure, although local dialogue structure is probably less affected.

sational games (Power, 1979) has been applied to the Map Task – see page 112. Other work includes standardising dialogue coding (Carletta *et al.*, 1995; Carletta *et al.*, 1997b) and examining disfluencies in dialogue speech (Bard & Lickley, 1997; Lickley & Bard, 1996).

3.3 A baseline system

To examine the contribution of the proposed method to recognition accuracy, we first need a baseline system. Having reviewed some typical state–of–the–art speech recognition systems, the chosen system can be designed using similar technology: Hidden Markov Models and a bigram language model. The acoustic models are the same for both baseline system and the system using intonation and dialogue constraints. The vocabulary size for the DCIEM task[3] is around 900 which is small to medium in speech recognition terms. The task is spontaneous dialogue speech. There are only a few systems with similar parameters for comparison, for example (Suhm & Waibel, 1994) or the speech recognition component of spoken dialogue systems such as Verbmobil (Wahlster, 1993) or TRAINS (Allen *et al.*, 1995; Ferguson *et al.*, 1996).

3.3.1 HMMs

Typical medium vocabulary speech recognition systems (e.g. for RM, (Woodland & Young, 1993)) use phone-based HMMs. Here, tied-state cross-word Gaussian-mixture density triphone models were used. That is, each model is context-sensitive one phone to both left and right, even across words, and the output probability density functions (pdfs) are mixtures of 8 Gaussian pdfs. Each model has three states. States are tied (shared among a set of models) according to a data-driven clustering technique, as provided by HTK (Young *et al.*, 1996).

[3]For the subset of dialogues used here.

This technique is decision tree based, and uses rules based on phonetic context. This means that models can be made even for triphones with no examples in the training data.

3.3.2 Language model

For well trained HMMs, the limiting factor in recognition accuracy is the language model. The baseline system should use the best language model possible, given the limited amount of training data available.

- ***Data***

The DCIEM corpus is described in section 6.3. The data was still being labelled as work progressed, so two data sets were used (named 4 and 5 for historical reasons): set 4 was an initial development set of 20 dialogues; set 5 is a larger set of 50 dialogues, which includes all of set 4. For language model estimation, all training data could be used, since word level transcriptions were available. The test set of 5 dialogues was constant throughout. The training and testing portions were selected so that no speaker appeared in both; this is true for the HMMs, dialogue model and intonation components also. Therefore, the whole system is truly speaker independent

- ***Dealing with non-words***

Because the DCIEM corpus speech is spontaneous, there are non-speech sounds and disfluencies to deal with. Many disfluencies consist of aborted and repeated words, such as "to the sou...the north". Although these are often fragments of real words, and are transcribed in the training data, dealing with these problems in the language model is not straightforward. It has been noted (Lickley & Bard, 1996; Bard & Lickley, 1997), that human listeners largely ignore, or in fact appear not to hear, disfluencies. In other words, some repair process is taking place. Here,

I take a very simplistic approach to the problem: non-words are grouped into a class which is called NW. This word appears in both the language model and the dictionary – where it is pronounced "sil" (that is, silence, which is modelled by a 3-state HMM). Other special words are used for filled pauses and aborted words, as shown in table 3.1; they are all pronounced "sil". This makes sense, because, for the data we are using, the segments labelled as "sil" actually contain a variety of background noises – I am simply extending the silence class to cover other non-speech sounds. Of course, aborted words *are* speech, but not speech we want to recognise.

AB	aborted word
FP	filled pause
NW	non-word, including non-speech sounds and silence

Table 3.1: Special words

Other words particular to conversational speech, such as "Uh-huh" are treated as normal lexical items and given full pronunciations. The problems of annotating disfluent speech were considered in section 4.5.1. Having the three homophones AB, FP and NW as separate words is not necessarily optimal. They could have been grouped as a single word as far as this work was concerned. However, they do have distinct linguistic functions and syntactic properties: aborted words and filled pauses don't typically end sentences; filled pauses can indicate the speaker's desire to continue to speak, for example "NW okay FP you've travelled east...". The treatment of disfluencies here is crude, to say the least, but work on disfluency in automatic speech recognition is a subject of research itself, and to attempt a more complex treatment here would be too ambitious. As far as assessing recogniser performance is concerned, the special words in table 3.1, plus Hm, Huh, Uh and UhHuh, are ignored. This does not artificially improve results - including these words actually gives slightly higher word accuracies.

- ***What is the "best" language model ?***

As I explain in section 4.2.1, perplexity is a good criterion for selecting a language model. The model with the lowest perplexity on a *held-out* portion of the training set was chosen.

- ***Training the language model***

Bigram language models were trained using the CMU-Cambridge Statistical Language Modeling [sic] Toolkit (Rosenfeld & Clarkson, 1997) and locally developed software. The amount of data available is shown in table 3.2.

Data set	dialogues	moves	words
set 4	20	4k	24k
set 5	50	10k	66k*
test	5	1k	7k

Note: 1k = 1000

* *Of which 55k are in sentences containing only set 4 vocabulary words*

Table 3.2: Data set sizes

- *Vocabulary mismatch*

Initial experiments used data set 4. This, plus the test set, has a vocabulary of around 900 words – I will call this the set 4 vocabulary. As more data was labelled, set 5 became available, and the vocabulary size increased to 1200. The increase can be attributed to two factors: the new dialogues are for different maps with new entities; the number of speakers increased. The test set had been fixed at the start and was covered by the 900 word vocabulary. Therefore, when training language models using set 5 data, there were two choices. Either the vocabulary could be extended so that the entire training set could be utilised, or only that part of set 5 which contained only set 4 vocabulary words could be used. The second option

Test set perplexity		Training data set 4	Training data set 5
Language model vocabulary	set 4	27.6	23.6
	set 5	n/a	23.1

Table 3.3: Effect of choice of vocabulary on test set perplexity

was chosen so that results for the new data were directly comparable to those using language models trained on set 4. This result shows the effect of training data set size on language model perplexity. When training language models on set 5 data, only those sentences which consisted entirely of set 4 vocabulary words were used – this reduces the number of usable training tokens (words) from set 5 by 17% to 55k[4]. The size of each of these data sets is shown in table 3.2.

Table 3.3 shows the effect of vocabulary choice and training set size on the perplexity of a backed-off bigram model – this type of model will be described on page 58 in chapter 4. The perplexities for models trained on all of set 5, and those set 5 sentences with set 4 vocabulary, are similar: 23.1 and 23.6 respectively. The effect of reducing the amount of training data is mitigated by the vocabulary size reduction (from 1200 to 900).

- *Training set size*

Some initial experiments simulated the lack of data for the move type-specific models (see later) by training the "general" model on only a fraction (1/12th, because there are 12 move type-specific models) of the training data. These experiments showed the effect of the lack of data well, but since the aim was to improve word accuracy over a baseline, a model trained on the entire training set was used in the baseline system. The perplexity of the language model as a function of training set size is given in table 3.4 and further illustrated in figure 7.4 on page 148. The vocabulary is set 4 in this case.

[4]1k = 1000 words

Training set	Test set perplexity
1/12th of set 4	43.1
all of set 4	27.6
set 5 (set 4 vocabulary part only)	23.6

Table 3.4: The effect of training data set size on back-off bigram language model perplexity. Vocabulary is set 4.

- *Optimising the language model*

Various choices can be made when training the LM, such as the back-off threshold value (see page 58), the method of calculating the back-off discounts and the vocabulary used. As everywhere, perplexities quoted here are for the test set, but perplexity on a held-out portion of the training set was the only information actually used to choose the best model. The choice of backing-off method has a small effect on test set perplexity, as does the cutoff at which backing-off begins. The cutoff is one less than the minimum count of bigrams required to include a bigram probability in the language model. Bigrams with counts at or below the cutoff are backed-off. A cutoff of 0 means that all bigrams occurring in the training data have probabilities estimated, and non-occurring ones are backed-off; a cutoff of 1 means that bigrams occurring only once or not at all will be backed-off. Experiments showed that, for this task, the lowest perplexity[5] model was a backed-off bigram, with discounts computed using the Witten-Bell (see page 61) method with a cutoff of 0. The perplexity for this type of model, trained on all set 5 training data which has a set 4 vocabulary, is 23.6 on the test set.

Finally, to demonstrate how much structure even a bigram model captures, table 3.5 gives test set perplexities for the best backed-off bigram model and a unigram model both trained on set 5 with set 5 vocabularies.

The vocabulary size of around 900 means that there are potentially 810 000 bigram probabilities to estimate. By backing-off many of these to unigrams, only

[5]On a held-out portion of the training set

Language model	Test set perplexity
unigram	110
backed-off bigram	23.1

Table 3.5: Comparison of unigram and bigram language models – set 5 training data and set 5 vocabulary

around 10 000 parameters (unigram and bigram probabilities – back-off weights are not free parameters) remain in the best language model. This relatively compact language model can be reliably estimated from data.

3.3.3 Results

The lowest perplexity backed-off bigram with a set 4 vocabulary and trained on set 5 was used in a speech recogniser using the HMMs described above. The accuracy of the system is calculated as:

$$\text{Accuracy} = \frac{\text{correct} - \text{insertions}}{\text{total number of labels}} \times 100\%$$

where the total number of labels is for the correct transcription and the number of correct and inserted labels was computed by a Viterbi alignment of recogniser output and correct transcription using HTK. The word error rate (calculated as 100 - Accuracy) was **24.8%**. This compares favourably with other systems, as summarised in table 3.6. The two comparable systems are reviewed on pages 64 and 72. We can see that the baseline system achieves state-of-the-art performance, and that it therefore provides a fair benchmark for the new method which is described in this thesis.

System	Vocabulary size	Language model perplexity	Word error rate
Suhm & Waibel (1994)	1200	35	34%
Eckert *et al.* (1996)	1500	20	25%
This system	900	23.6	24.8%

Table 3.6: How the baseline system performance compares with others

Chapter 4

Language modelling

- ***What does a language model do?***

In the speech–recognition–as–search paradigm introduced on page 1, the language model is a *constraint*. It imposes restrictions on the search space, leading to better solutions in less time.

We need to estimate the likelihood that a given sentence is in the language. In practice, we also need to be able to do this for partial sentences – that is, *any* word string. Because language can only really be learned from observation, and because we can never observe enough natural language to have seen all possible sentences, this estimate will be an estimate of how frequent the sentence is in a (hopefully) representative example of the language. This example is called a training corpus.

A wide selection of language models is available, ranging from hand-crafted grammars with thousands of rules to statistical models with parameters estimated from large corpora. Only a small subset of these is useful for speech recognition. To describe language modelling, a few special terms will be used, and these are defined in the glossary below.

Language Modelling Glossary

language	a subset of all possible word sequences
grammar	a perfect description of the language
coverage	how much of the language the model can generate
overgeneration	when a model generates sentences *not* in the language
perplexity	a measure of the complexity of the language, defined on page 49; also a measure of how well a model matches a test corpus
robust	the model parameters are well estimated
fragile	the opposite of robust

For probabilistic models, which can estimate the likelihood that *any* given word string would occur in the language, coverage comes to mean how well the model accounts for word strings which are likely to occur, but did not do so in the training corpus. Models which estimate that such strings *never* occur, just because there were no training examples, have poor coverage (and consequently high perplexity[1]). For probabilistic models, overgeneration is not relevant, since perplexity measures how well the model matches the language.

To quote Meteer & Iyer (1996): "The interesting problem in language modelling is how to bring generalisations above the level of the words themselves to the text." This problem is taken into consideration in this chapter, where a selection of language models are described. The problem of generalisation is addressed through word class systems, parameter smoothing and backing off and utterance type specific language modelling.

[1]In fact, any model which estimates zero probability for a particular word string will have infinite perplexity over any test data which contains that string.

- ***Requirements for speech recognition***

To facilitate speech recognition in reasonable time, we must impose some restrictions on the language model we choose. It is highly desirable that the language model be convertible to a finite state network, which means that only left context may be used to condition word probabilities. Models which are not readily represented as finite state networks are not easily integrated with the acoustic models into the decoding algorithm – (Rabiner & Juang, 1993, page 448), amongst others. Restrictions will also result from the limited number of observations of the language available – that is, the size of the training corpus.

4.1 Classes of language model

4.1.1 Stochastic context-free grammars

Stochastic context-free grammars (SCFGs) are simply context-free grammars (CFGs) with probabilities for each production. These are "top-down" or generative models, are well understood, and the probabilities are trainable from data. There are typically far fewer parameters in a SCFG than a word N-gram model. This is because the *structure* of the SCFG is predetermined since the number of nonterminals is fixed (the number of terminals being given by the vocabulary size); only the probabilities required in the rules are learned from data. The reduction in free parameters is highly desirable since it allows more robust parameter estimation for a given amount of training data. On the other hand, the fixed structure results in poor coverage - the model cannot learn to make new productions. SCFGs are not suitable for direct use in speech recognition (Rabiner & Juang, 1993), but can be converted to N-gram models using techniques such as that described in (Stolke & Segal, 1994) – see page 58.

4.1.2 Left context-dependent models

The alternative to top-down generative models like SCFGs is a bottom up approach. Here, no parsing is done since no overall structure is modelled. Rather, word sequences are assigned probabilities directly.

- ***Word pair***

The simplest form of left-context dependence is a word-pair language model which simply consists of a list of valid two-word sequences. All valid pairs are equiprobable; all other pairs are impossible. For some tasks, such as the Resource Management task (RM) (Linguistic Data Consortium, 1996b), a simple word pair model can perform surprisingly well. When the task has a very "rigid" grammar where the set of possible sentences is basically fixed, and no other sentences must be allowed (which is the case in the RM), word-pair models have sufficient *coverage* with low *perplexity*. They have no probabilities to estimate, so do not require large training corpora.

- ***N-gram models***

If we add probabilities to a word pair model, so that words follow other words with differing probabilities, we get a *bigram* model. Typically, all words will be allowed to follow all other words, some with much higher probabilities than others. If we then condition the probability of a word not just on the immediately preceding word, but on the preceding $N-1$ words, we get an N-gram model:

$$P(w_1, w_2, w_3, \ldots w_M) = \prod_{i=1}^{M} P(w_i | w_{i-N+1}, \ldots w_{i-2}, w_{i-1})$$

where we typically estimate $P(w_i|w_{i-N+1}, \ldots w_{i-2}, w_{i-1})$ from data:

$$\tilde{P}(w_i|w_{i-N+1}, \ldots w_{i-2}, w_{i-1}) = \frac{C(w_{i-N+1}, \ldots w_{i-2}, w_{i-1}, w_i)}{C(w_{i-N+1}, \ldots w_{i-2}, w_{i-1})} \quad (4.1)$$

where $C(\cdot)$ is a counting function applied to some corpus of training data and $\tilde{P}$ is an estimate of P.

- ***Longer span models***

Training data requirements generally limit N in N-gram models to 2 or 3 (Rabiner & Juang, 1993, page 447), so we cannot simply model longer term effects by increasing N. This is because the number of N-grams is $(N_V)^N$. One of the most striking long term effects is known as *word recurrence* – words used recently are more likely to be used again. One method for modelling *word recurrence* is to use a *cache*; this is described in section 4.4.3. The most common technique for modelling longer span dependency is to mix N-gram models of different N. This is known as *backing off* and is described on page 58.

- ***Finite state models***

It is possible to generate a finite state model directly. For small tasks (for example, recognising telephone numbers) this can be done by hand. This has the great advantage of precise coverage without overgeneration. Manual generation of such models is of course very tedious, and not practicable for most real-world tasks. Some attempts have been made to combine the precise coverage of finite state models with the probabilistic nature of N-gram models (Eckert *et al.*, 1996).

4.2 Meeting the requirements

4.2.1 Desirable properties

To select amongst those language models meeting the restrictions for use in speech recognition, we need some criteria.

- ***Probabilistic models***

The language model will be used to compute the probability of word sequences[2]. This will be useful in ranking a list of candidate word sequences in order of likelihood, ordering and pruning the search, and for ranking the likelihoods of a particular word sequence for each of a set of language models. An application of the latter would be utterance type classification based on word sequence, using a set of utterance-type specific language models. Therefore, probabilistic language models are better than non-probabilistic ones.

- ***Models with low perplexity***

Perplexity is a measure of the average "branching factor", or the typical number of words which can follow a given word sequence. For speech recognition, fewer possible words means an easier task for the recogniser. So, a language model with low perplexity is more constrained, and will generally result in faster and more accurate recognition. The relationship between perplexity and word accuracy is not guaranteed, although we *expect* models with lower perplexity to produce better word accuracy.

Consider first a simple language model in which all of the N_V words in the vocabulary are allowed to follow any other word (with probability $1/N_V$). The perplexity of this model is N_V . Perplexity is defined for probabilistic models too, where the probability of words following each other is not uniformly $1/N_V$.

[2]That is, the probability of them occurring in the language.

Perplexity, B, is defined in terms of entropy, H. In practice, we can only estimate probabilities using some test data, and thus only an estimate of perplexity can be obtained. The more data used to train and test the model, the better this estimate should be. The test data should be a held-out set, as explained below. From Rabiner & Juang (1993), page 450:

$$B = 2^H$$

and we estimate H to be H_p over Q words of data:

$$H_p = -\frac{1}{Q} \log P(w_1, w_2, \ldots w_Q)$$

which for an N-gram model is

$$H_p = -\frac{1}{Q} \sum_{i=1}^{Q} \log P(w_i | w_{i-1}, w_{i-2}, \ldots w_{i-N+1}) \tag{4.2}$$

- *Held-out method*

A test data set is required to measure language model perplexity. We cannot use the actual test set[3], so a *held-out* scheme is used in which some part of the training data is *held out*, that is, it is not used for estimating the model parameters. The held-out data can then be used for estimating the model perplexity. Simply using the same data to estimate the model parameters *and* perplexity would give misleading results. Perplexity will generally be lower over training data than test data. Optimising the model perplexity over the training set would therefore lead to a language model "tuned" to the training set and *less* likely to have low perplexity over test data. In practice, the held-out method is used for optimising

[3]That would be cheating !

the estimation *process* – selecting the length of N-grams, the type of discounting when backing-off and so on. A model is then re-estimated using this method from the *whole* training set. This avoids wasting valuable training data. The held-out part of the training set (typically some fraction, such as one third) can be rotated – if the training data is split three ways, then three held-out sets can be used to get three different estimates of the best process[4].

4.2.2 Models with desirable properties

The selected model must meet the restrictions given at the start of this chapter, and this means that we must be able to represent it by a finite state network. If this is not possible, a method for approximate conversion to a finite state network must be used.

- ***N-gram models***

Because N-gram models are easily represented as finite state networks, those language models which fail to meet the requirements on page 45 can still be used via estimation of a N-gram model. For example Stolke and Segal (1994) use a SCFG to compute N-gram probabilities – see page 57.

[4]This technique is sometimes called cross-validation. I will not use this term because it means different things to different people.

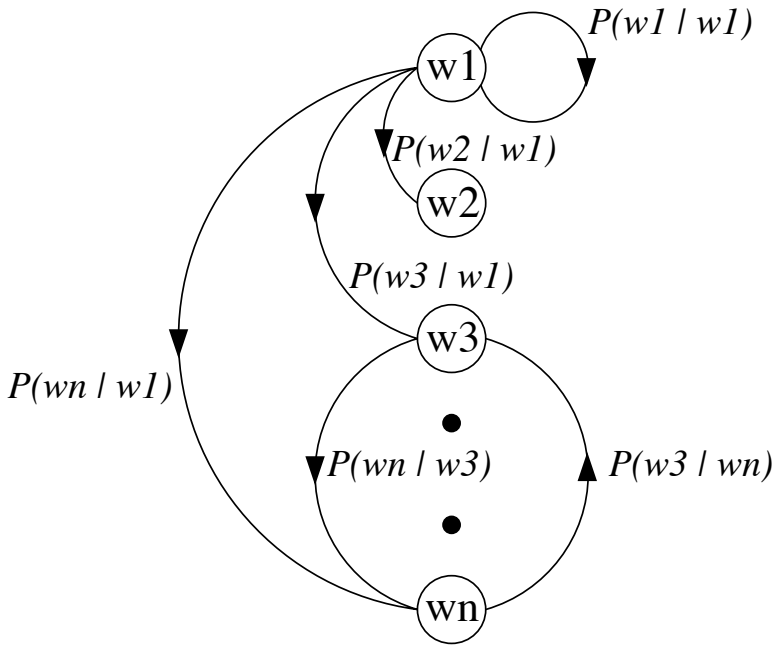


Figure 4.1: Representation of an bigram model by a finite state network (not all arcs are shown)

- *Representation as finite state network*

Figure 4.1 shows an example for N=2, where nodes represent words and arcs represent probabilities. In general, nodes represent (N-1)-tuples of words – that is the conditions in the conditional probabilities (e.g. $\{a, b\}$ in $P(c|a, b)$). For example, in a trigram, each node is labelled with an ordered pair of words; we can see that the size of the network will increase exponentially with N – there are $(N_V)^{(N-1)}$ nodes and $(N_V)^N$ arcs where N_V is the vocabulary size.

In the figure, not all arcs are shown, and the start node is not made explicit. In a practical network, we would have two special nodes (perhaps labelled !ENTER and !EXIT), and constrain the path through the network to start at one node and finish at the other. In this way we also model the probabilities of words (or more generally, word (N-1)-tuples) starting and ending sentences.

4.3 N-gram language models

The baseline recogniser and the new method both require language models (the new method requires several models). All these models perform the same task: they constrain the search during recognition. From the literature review above, it is clear that N-gram language models are the obvious choice. The reasons can be summarised thus:

- easy to integrate into the recogniser
- allows parameter smoothing:
  - can have non-zero probability for zero-frequency N-grams
  - very simple to interpolate between two models
- wide variety of training techniques to:
  - compensate for sparse data
  - minimise perplexity
  - optimise number of free parameters

4.3.1 Robustness

Inevitably, language models are trained on minimal amounts of data; if more data is available, models are more likely to be made more complex (for N-gram models, this means large values of N) than to be better trained! A lack of training data leads to fragile models: they will have poor coverage and badly estimated probabilities. A variety of techniques is available to compensate for these problems, all of which work by adjusting the raw estimates of the model probabilities given by equation 4.1.

- ***The zeroton problem***

One major problem with probabilistic models such as N-grams, is how to estimate the probability of events *never* seen in the training data. There will always be such events (N-grams in this case) for two reasons: 1) some N-grams really *don't* ever occur in natural language; 2) there was not enough training data. There is a similar problem with all rare events in the training data. If we saw an event just once in a corpus, this is not a reliable estimate of its "true" frequency. The simplest solution is to use a "floor" probability, so that the probability of an N-gram never falls below some small amount (equivalently, no N-gram has a frequency below a certain value). Fixed floor probabilities are crude and must be manually selected. A more general solution is to smooth the frequency counts, which adjusts both zero and non-zero frequencies.

4.3.2 Smoothing

Smoothing the parameters of the language model is an attempt to "iron-out" irregularities in the frequency of frequencies distribution – for example by smoothing the distribution by fitting a function. An example frequency of frequencies distribution is given in table 4.1.

frequency of N-gram	number of different bigrams (N=2) with that frequency
0	711601
1	5074
2	1445
3	669
4	385
5	247
6	210
...	

Table 4.1: A frequency of frequencies table for the DCIEM training set 5, set 4 vocabulary.

Table 4.1 is for the training data used for the baseline language model – DCIEM set 5 with set 4 vocabulary. Only around 10% of the possible bigrams are found in the training set. The table is truncated at a frequency of 6.

In a simple N-gram model, the probabilities are estimated by the Maximum Likelihood Estimation (MLE) defined by equation 4.1. Using this estimate, the probability of events never seen in the training data is exactly zero. We would like to be able to replace that zero with a better estimate. To do that, we will have to remove some of the "probability mass" from the non-zero frequencies and assign it to zero-frequency events.

If C is the total number of N-gram *observations* seen in the training data, and C_r is the number of distinct N-grams seen r times, then

$$C = \sum_{r=0}^{\infty} C_r r$$

Now if $r^\star$ is the smoothed frequency which replaces frequency r, we must ensure that $\sum_{r=0}^{\infty} C_{r^\star} r^\star = C$ and since $C_{r^\star} \equiv C_r$ then $\sum_{r=0}^{\infty} C_r r^\star = C$

- ***The Good-Turing method***

The Good-Turing method (a good description of which can be found in (Church & Gale, 1991), for example) for smoothing the frequency-of-frequencies distribution is:

$$r^\star = \frac{(r+1)C_{r+1}}{C_r}$$

We must preserve the property $\sum_{r=0}^{\infty} C_r r^\star = C$. From the above, $C_r r^\star = (r+1)C_{r+1}$, so $\sum_{r=0}^{\infty} C_r r^\star = \sum_{r=0}^{\infty}(r+1)C_{r+1}$. Now, $C = \sum_{r=0}^{\infty} C_r r$, but for $r = 0$, $C_r r = 0$ so the lower limit on the sum can be replaced by $r = 1$. A simple substitution of $(r+1)$ by r, so that $(r+1) = 1$ becomes $r = 0$, shows that $\sum_{r=0}^{\infty} C_r r^\star = C$.

r	$r^\star$
0	0.00072
1	0.57
2	1.4
3	2.3
4	3.2
...	

Table 4.2: Good-Turing smoothing of the frequencies from table 4.1.

Table 4.2 shows the effect of Good-Turing smoothing on the frequency distribution from table 4.1 using the equation above. Enhanced versions of this method are possible by, for example, smoothing the C_r first, or taking unigram counts into account when estimating bigram frequencies (Church & Gale, 1991). The Good-Turing method can be applied to calculating discounts in a backing off scheme as described in section 4.3.3.

- ***Interpolation***

Often, we are estimating a language model for a specific domain where the amount of training data is limited. However, there may be a large amount of data available from another domain. A language model trained on the larger corpus will be more robustly estimated, but will not match the target domain. A simple solution to this would be to interpolate the parameters of the domain-specific (but fragile) model with this model in the hope of getting the robustness and coverage of the large-corpus trained model but with the domain-specificity of the fragile model.

The "domain" can be defined in various ways. One interesting definition is that *each utterance* belongs to a particular domain. There is a set of domains reflecting, say, possible utterance types. Now, consecutive utterances in a corpus need not belong to the same domain – the domains can be "interleaved". Each domain contains many utterances. Now, in estimating a language model for each of these domains, there is probably not going to be enough in-domain data for

training. Therefore, the model parameters could be interpolated with a more robustly estimated model – and the obvious choice is a *domain independent* model, which is simply a model trained on data from *all* domains. This is a key idea[5] in the work here, and the experimental details are described in section 4.6.4.

- *Estimation–maximisation model interpolation*

Two or more language models can be interpolated such that the probability of a given word string assigned by the interpolated model is simply a weighted sum of the probabilities assigned by the original models:

$$P(\text{word string}) = \sum_i \lambda_i P_i(\text{word string})$$

The weights λ_i can be found using the estimation maximisation (EM) algorithm[6], which chooses weights which minimise the perplexity of the interpolated model over some held-out[7] training data. This algorithm is implemented in the CMU language modelling toolkit (Rosenfeld & Clarkson, 1997) which was used to estimate the (word level) language models in this thesis. Experimental results are described on page 76. Linear interpolation of language models has attractive properties – pointed out in (Rosenfeld, 1994): it cannot hurt, since the EM algorithm guarantees not to increase perplexity (on the held-out data, at least); very little held-out data is required to compute the weights; the weights do not need to be specified very accurately.

[5]Interpolation of language models is not a new idea.
[6]Also known as the expectation–maximisation algorithm.
[7]See page 49 for a description of the held-out method.

- ***Other methods***

  - *Cooccurence smoothing*

Essen & Steinbiss (1992) extend a technique originally applied to HMM parameter smoothing to smoothing the probabilities of stochastic language models. This technique makes use of the observation that some words frequently occur in the same context as other words (for example, most nouns can occur in the context /the ___/). This property can be used to smooth the probabilities in, say, an N-gram language model. A confusion matrix can be computed, containing the probability of two words occurring in the same context (their *confusibility*). In the case of a word bigram model, the confusibility is used to compute a smoothed conditional probability for a bigram based on a weighted sum of conditional probabilities for all confusible words. The weights are the values from the confusion matrix. Thus the smoothed conditional probability of a bigram depends on the conditional probabilities of confusible bigrams. Variants are possible where the smoothing takes place not over the current word (the one being predicted) but the preceding word, or indeed both. Perplexity reductions of up to 10–15% are shown using this technique.

  - *Cooccurence with backing off*

Pereira *et al.* (1996) use cooccurence to distribute the discounted probability mass in the back-off method (Katz, 1987) (see also section 4.3.3). The usual method for redistribution is based on the observed (N-1)-gram prefix frequency. The cooccurence method uses the frequency distribution of "similar" words. The effect of this is to average together the standard back-off probability estimates for a group of similar words. Similarity in this case is defined as the *Kullback-Liebler* distance, which is a relative entropy measure. Perplexity reductions obtained (on the Wall Street Journal task) by this technique are small, at only 2.4%, and the

corresponding word error rate reduction reported is 21.4% to 20.9% – a reduction of 2.3% (relative).

- *Using stochastic context-free grammars*

Although stochastic context-free grammars (SCFGs), as mentioned in section 4.1.1, are powerful models with typically far fewer parameters than N-gram models, they are not directly suited to use in speech recognition. Stolke & Segal (1994) give a method for generating N-gram probabilities from SCFGs, thus getting what they call "the best of both worlds". However, best results are obtained by smoothing the N-gram probability estimates obtained by this method with directly estimated ones, because SCFGs suffer from limited coverage (as defined on page 43).

4.3.3 Backed-off N-gram models

To model longer term dependencies, we would like N to be as large as possible. However, as the number of parameters in a straightforward N-gram model is the number of items in the vocabulary raised to the power N, there is unlikely to ever be enough data for large N. We know that, as N increases, the number of different N-grams actually found in data becomes a smaller and smaller fraction of the total number of possible N-grams. Since we only need to reliably estimate probabilities for relatively common N-grams, we can make use of this property.

Consider a trigram (N = 3): {a, b, c}. It contains the shorter term: {b, c}. We wish to estimate $P(c|a,b)$ but have not seen {a, b, c} enough times in data to be able to do that reliably. We can, however, estimate $P(c|b)$ reliably. We can base our estimate of $P(c|a,b)$ on $P(c|b)$ through a technique known as *backing off* (Katz, 1987).

Let

$$P(c|a,b) = \alpha.P(c|b) \text{ if } C(a,b,c) \text{ is below some threshold}$$

where $C(\cdot)$ is the counting function from before, and α is some weight to ensure that probabilities for a given word history sum to 1, that is:

$$\sum_{w \in V} P(w|a,b) = 1 \ \forall\, a,b \quad \text{where } V \text{ is the vocabulary } a,b,c\ldots$$

Clearly, if we estimate probabilities with equation 4.1, then α will be zero. Therefore, we need to *discount* some of the *probability mass* from those N-grams with non-zero frequency so that $\alpha > 0$. Generalising, from (Katz, 1987):

$$P(w_N|w_1^{N-1}) = \begin{cases} \tilde{P}(w_N|w_1^{N-1}) & \text{if } C(w_1^N) > k \\ \alpha_{w_1}^{w_N-1} \cdot \tilde{P}(w_N|w_2^{N-1}) & \text{otherwise} \end{cases} \tag{4.3}$$

where w_1^N is the sequence of words $w_1, w_2, \ldots w_N$, k is a threshold and $\tilde{P}(\cdot)$ is now estimated using:

$$\tilde{P}(w_N|w_1^{N-1}) = \frac{C(w_1, w_2, \ldots w_N) - d(C(w_1, w_2, \ldots w_N))}{C(w_1, w_2, \ldots w_{N-1})}$$

where $d(\cdot)$ is a *discounting* function which removes probability mass from higher frequencies so allowing it to be redistributed to lower ones. In Katz's notation, w_N is the word whose probability is being estimated and w_1 is the first word in the N-gram. By adjusting the threshold k, the number of N-gram probabilities that are *backed-off* to weighted (N-1)-gram probabilities can be controlled. All backed-off N-grams with the same N-1 initial items share the same back-off weight α, so high cut offs result in fewer parameters to estimate.

- ***Discounting functions***

There are a variety of discounting functions – $d(\cdot)$ – to choose from.

- *Fixed discounting*

In the simplest discounting scheme (Ney *et al.*, 1994), a fixed discount is removed from all frequencies greater than the threshold k. Typically $d = 0.5$.

- *Linear discounting*

Now the discount is proportional to the frequency - a fraction of the frequency count is discounted (Ney *et al.*, 1994): $d(r) \propto r$.

- *Good Turing method*

Katz's (1987) original suggestion was to use a Good Turing method, as applied to language modelling (see (Church & Gale, 1991), amongst others) to compute a frequency for unseen N-grams, based on the frequency of frequencies distribution. As in section 4.3.2, the frequency of frequencies distribution is modified (see table 4.1 for an example). Zero frequencies are modified to a small non-zero frequency. From this frequency we can see what the total probability of all the "zerotons" (unseen N-grams) should be.

In the simple smoothing of section 4.3.2, all zerotons would now be assigned a uniform (small) probability based on the modified zero frequency entry in the table. In a backing off scheme, this probability is distributed amongst the zerotons according to the (N-1)-gram frequency, according to equation 4.3. Variations of this method are available depending on how far down the frequency of frequencies distribution the Good Turing method is applied.

- *Witten-Bell discounting*

This method is from (Witten & Bell, 1991), as implemented in (Rosenfeld & Clarkson, 1997), from which the following is taken:

> The discounting ratio is not dependent on the event's count, but on t, the number of types which followed the particular context. It defines $d(r,t) = n/(n+t)$, where n is the size of the training set in words. This is equivalent to setting $P(w|h) = c/(n+t)$ (where w is a word, h is the history and c is the number of occurrences of w in the context h), for events that have been seen, and $P(w|h) = t/(n+t)$ for unseen events.

- ***Which method is best?***

We have seen that there is a variety of methods for estimating robust N-gram models, but which is the best one? This is easy - it is the method which gives the model with the lowest perplexity on held-out data. In other words, the best method can only be determined experimentally, and this is what was done. The experiments are described in section 4.6.4.

- ***Representation as finite state network***

Like any N-gram model, a backed-off N-gram model can be represented as a finite state network. This is most easily illustrated by figure 4.2 on page 62. As the figure shows, some transitions share the same value (unigram probabilities) – the number of parameters in the backed-off model is typically far fewer than in a "full" N-gram model.

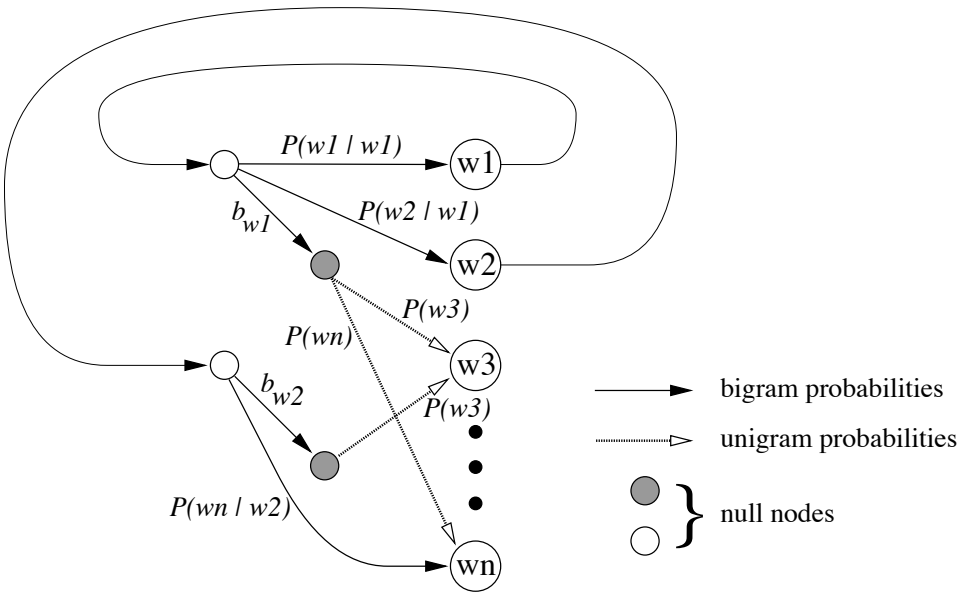


Figure 4.2: Representation of a back-off bigram model by a finite state network (not all arcs are shown)

4.4 Adaptation

A language model is trained on a corpus of data, and used to model some test data. There will always be a degree of mismatch between the training and testing data[8], which can be for a number of reasons: the training data could be from a different domain to the test data, about a different topic, from a different speaker or set of speakers, and so on. To mitigate this mismatch, the model can be adapted to the test data. I will first explain what I mean by *adaptation* and then consider how it might be achieved. In the literature, adaptation is taken to mean changing the parameters of a language model during recognition – I will call this *on-line* adaptation. I will use the term "adaptation" more generally to mean modification of model parameters after their initial estimation from training data.

[8]Perplexities of language models are generally greater on testing data than on training data.

4.4.1 Adaptation to what ?

The causes of mismatch listed above are all motivations for adapting the model, and each will determine what is adapted, and when.

- ***Domain***

The most obvious problem we might want to overcome is that the model was trained on data from a different domain. This could be because not enough data from the target domain was available for training, or because we do not know precisely what the target domain is.

- ***Speaker***

Different people speak and write in differing ways, and this can cause mismatch between the language model and the speech being recognised.

- ***Topic***

Adaptation to topic is usually applied to recognition of longer passages, where use can be made of effects such as word recurrence (see page 47). Topic can be seen as similar to domain, but with more localised effects on the language – entities mentioned may be topic dependent, and so there are strong effects on unigram frequencies in particular. Adaptation to topic is a case where adaptation of only some model parameters might be desirable – see below.

- ***Dialogue act***

All adaptation considered so far has been either relatively long-term, or incremental. Long term means that the language model remains fixed from utterance to utterance, with parameter adaptation taking place between longer passages,

or session of use. Incremental means that the model parameters do not change radically from one utterance to the next.

There are striking differences in grammar between sentences, and we can take advantage of this. What we would like to be able to do is adapt the language model to each unknown utterance. To do this, we need to measure some property of the unknown utterance to base the adaptation on.

Such a property is the *dialogue act*, which encodes the rôle of the utterance in the dialogue. I will use instead the term *utterance type*. This property can be hand labelled by examination of a dialogue transcript. Utterance type encodes the *rôle* of the utterance in the dialogue, and this is reflected strongly in the surface form (word sequence). In order to detect the type of an utterance, some *cues* are required.

Eckert *et al.* (1996) use *dialogue step* dependent language models – this is a very similar approach to the one taken here, although the dialogue step system is much simpler than that of conversational games, with only five steps being defined for train enquiry task: initial, time, goal-city, source-city and date. A dialogue step independent model was used in parallel with these five models, to enhance coverage. Up to 6% word error rate reductions are reported. Vocabulary size is around 1500 words, language model perplexities are around 20 and word error rates around 25%, all of which are very similar to the work in this thesis.

- *Acoustic cues*

Considering utterances in isolation, all the cues we have (assuming we are not using additional data such as vision) are in the acoustic signal. Ideally, we would like to use cues which can be extracted prior to doing speech recognition – although a two pass approach is possible. Kowtko (1996) has examined the function of intonation in task oriented dialogue – this is reviewed on page 93.

- *Contextual cues*

In dialogue situations, as seen in section 6.2.1, the previous utterance(s) give strong cues to the current utterance type. This can be exploited by a dialogue model, which is discussed in detail in section 6.3.3.

4.4.2 Adaptation of what ?

Now that we have established cues for language model adaptation, what form is this adaptation going to take ? Here I will only consider probabilistic models (such as N-gram models) where the parameters are estimated from data.

- ***All parameters***

The simplest scheme would be to adapt all parameters – effectively selecting an entirely different model. For example, using utterance type as the property on which we are basing the adaptation, this means a particular (version of the) language model for each utterance type.

- ***Some parameters***

It is likely that some parameters of the language model will not need adapting. This is easy to see, for the case of adapting to utterance type, in the difference between statements and questions:

statement	You have a cottage
question	Have you a cottage ?

The inversion implies adaptation of probabilities for words likely to be inverted, but not for pairs like "a cottage". However, explicitly adapting only some parameters means identifying those parameters. This may be straightforward in a word-class based scheme, where we may elect only to adapt probabilities of certain classes: for example, the frequency of Map Task entities could be adapted

depending on the particular map in use. In non-word class system, it may be simpler to opt for adaptation of all parameters.

4.4.3 Implementation

For some types of adaptation, we can *"pre-adapt"* the language model, for example, to a new domain, with extra training material. In other cases, adaptation can only be made gradually as recognition proceeds – for example, adaptation to a previously unheard speaker. I will call these possibilities off- and on-line.

- ***Off-line***

Language model estimation is generally computationally intensive, and ideally carried out before recognition. Schemes which allow such off-line adaptation are therefore preferable, particularly as language models become more complex and vocabularies become larger.

- ***On-line***

When off-line adaptation is not possible, or not optimal, language model parameters can be changed as recognition proceeds. The cues used for adaptation must be present in the speech already processed. Typically this means the words recognised so far. One way to use this cue is with a cache-based model.

  - *Models with a cache*

One effect that can only be accounted for on-line is word recurrence: words which have occurred recently are more likely to occur again. This is typically (Clarkson & Robinson (1997) for example) modelled with a cache. A cache is simply a store of all recently occurred words. The probability of words in the cache is boosted.

As the distance (in words) back to the previous occurrence of a particular word increases, the amount of "boosting" is typically reduced.

Using a cache is effective in producing language model adaptation to various effects: domain, topic or speaker. However, at or near the start of a new recognition run, when the cache is empty or only contains a few words, the effect is either absent or very crude. Cache-based models may therefore be best suited to longer passages.

4.5 Models designed for conversational or dialogue speech

The grammar of spontaneous speech is clearly different to that for written language and the grammar of conversational or dialogue speech is obviously different from other spontaneous speech. The Switchboard corpus (Linguistic Data Consortium, 1993-7) is a good example of spontaneous dialogue speech. Meteer & Iyer (1996) provide a good review of the problems facing annotation and modelling of Switchboard data. They address the problems of disfluency and definition of the sentence in conversational speech, summarised below.

- *Disfluency*

One of the biggest problems associated with annotating and modelling spontaneous speech is that it contains a very high proportion of disfluencies. These range from simple effects such as repeating words, to longer term ones such as rephrasing entire sentences, or repairing mistakes long after they occur. For example, from the DCIEM Map Task corpus (Bard *et al.*, 1995):

> *"I .. now I have .. hmm, below the ruined monastery I have an overnight accommodation ... a little house."*

where, had they been asked to write it down, the speaker probably would have put"Below the ruined monastery, I have a little house."

4.5.1 Annotation issues

Meteer & Iyer (1996) describe the problems in annotating disfluent speech. Three categories of annotation problem are addressed: marking sentences; describing restarts; and non-sentence elements. In labelling such a large corpus, care must be taken to mark any information which may be of later use, since a second labelling pass is not possible because of the cost involved. Meteer & Iyer use the annotation scheme of Shriberg (1994) in which disfluencies are bracketed in such a way as to allow "repair" by deletion of the speech subsequently repaired by the speaker, leaving reasonably fluent speech. Filled pauses and other non-sentence elements are explicitly marked as such. For example, from (Meteer & Iyer, 1996):

transcription	Show me flights from Boston on uh from Denver on Monday
annotation	Show me flights [from Boston on + {F uh} from Denver on] Monday
repaired	Show me flights from Denver on Monday

In the repaired version, the section inside [...] is repaired by deletion of the portion before the +. The pause filler "uh" is deleted during a separate clean-up process.

In the labelling of the DCIEM Map Task corpus used in this thesis, only word level labels were used. Repaired or partial words are marked as such, along with the labeller's guess at what the full word would have been. No bracketing of repairs and so on was available. The word labels were cleaned up considerably before language models were built. This cleaning involved relabelling all partial words and non-speech as described on page 38. This is a very crude scheme, but greatly simplifies subsequent language modelling.

4.5.2 Dividing the input speech

One problem facing recognition of conversational speech is that of deciding where to divide the speech signal into chunks for processing. Smaller chunks are processed faster and may give some advantage for language modelling – see section 5.2.2. The definition of a sentence is not clear in spontaneous speech. Utterances are rarely grammatically well-formed, and there is typically a high proportion of sentence fragments and even sentences split across turns. Meteer & Iyer (1996) compare two segment hypothesising techniques: acoustic and linguistic. They conclude that a (manual) linguistic segmentation is more advantageous for language modelling (the models have lower perplexity) than an acoustic segmentation using pauses, silence, non-speech and turn taking. However, models trained on manually segmented data were not well matched to test data with automatically hypothesised segment boundaries.

In the DCIEM Map Task corpus, I did not attempt an automatic segmentation into utterances for either testing or training data, and used only the manually labelled *move* (see page 112 for a definition) units. Thus, mismatch between training and testing data is avoided. The problem of segmentation may be easier for this corpus, since there is a stronger dialogue structure than in the Switchboard conversations, so turn-taking is more explicitly marked by the speakers, both in the intonation and words spoken. Furthermore, these dialogues are goal oriented and take place between two cooperating participants, which means that the speakers try harder to mark turn taking, and are less likely to do things which do not help achieve the goal.

4.6 Sub-language models

In hand-crafted grammars for natural language, such as in (Alshawi, 1992), *sub-grammars* of often used. These are self-contained grammars which can be inserted into the main grammar. For example, we might write a grammar for telephone numbers, then treat "telephone number" as a single item in the main grammar. Such sub-grammars greatly simplify the task of writing the grammar, and in the case of probabilistic grammars, reduce the number of parameters.

With models of natural language estimated from data, something similar can be done. If common structures, like telephone numbers, can be found and treated as "sub-languages", better use can be made of the training data. The sub-language model only has to learn a simple grammar, and the number of items (lexical items plus sub-grammars) in the main grammar is reduced. The problem is of course: how do we define and detect these structures? In dialogue speech, there are natural divisions when the speakers exchange control or start and end turns, although this is confused by overlapping speech. These points are a starting point for defining a useful unit of speech.

4.6.1 Finding units in spontaneous speech

In section 4.5.2, dividing the input speech was motivated by both the need to process the input in reasonable sized chunks, and the finding that language models for linguistically segmented speech had lower perplexity then ones for un-segmented speech. The chunks are homogeneous – they are all of the same type, and the same language model is used for them all.

To use sub-grammars, each of which models chunks of differing types, the criteria for defining units is slightly different. As above, the unit must be something whose boundaries are easy to determine, and which is long enough to exhibit useful linguistic structure, yet short enough that there are a reasonable number of

examples in training data. Additionally, we require that the units form *clusters* in terms of grammar (and possibly other criteria, such as intonational tune; see chapter 5). Furthermore, it would be an advantage if sequences of these units exhibited some pattern so that cross-unit constraints can be used, as mentioned in section 2.1. The terminology used here is as follows: an utterance is a chunk of speech which contains just one of the units we will define; each utterance has a type, selected from a finite set of types.

- ***Turns***

Turns are a unit of dialogue in which a sub-goal of the dialogue is achieved. A turn typically consists of more than one utterance, and involves both speakers; for example, a question followed by a reply. The turn is a rather large unit, which is problematic for our purposes. Because more than one speaker is involved, there is structure *within* the turn which we may not be modelling well, and there will typically not be many turns per dialogue in the training data which will make modelling turn sequences difficult.

- ***Moves***

The theory of *conversational games* (Power, 1979) was introduced in section 2.2 for utterance type classification. For the reasons given there, the *move* is an ideal candidate for a unit of speech. Moves are smaller units than turns; there will be sufficient moves in the data to train dialogue models (see section 6.3.3). Furthermore, the set of move types form a natural set of classes which are motivated by dialogue theory and therefore, potentially, will form clusters in terms of language too.

Move types form a small set – 12 in (Kowtko *et al.*, 1993). Therefore there will hopefully be a reasonable number of examples of each type in the training corpus. As I will show, sub-language models for individual move types have lower

perplexity (on the same test data) than a model trained on utterances of all move types.

The set of move types in (Carletta *et al.*, 1995) strikes a balance between units of a convenient size and context independence. Smaller units (single phrases, for example) would be more context sensitive and larger units would require a larger set of types. As we saw in section 2.4.3, some of the move types in the set defined in (Carletta *et al.*, 1997b) *do* exhibit some context sensitivity – for example, the surface form of a reply-no move which is a response to a yes-no-question move will tend to be different than one which is a response to a move of another type (Hockey *et al.*, 1997). This could actually be exploited in a real system, because if we know the effect on surface form of the eliciting move, then we can reflect this in our language models and make the recognition task easier for ourselves (human or machine!) by asking questions in the right way – see page 106.

Suhm & Waibel (1994) use speech-act dependent word bigrams. The speech acts are identified by prediction of dialogue state (Ward & Young, 1993) and a semantic parser. The speech act categories include *give-info*, *suggest-time* and *interject*. The task is the English version of the Spontaneous Scheduling Task, a forerunner of Verbmobil (Wahlster, 1993). As mentioned already, Eckert *et al.* (1996) use *dialogue step* dependent language models, which is a similar system to that used in this thesis.

- ***Word phrases***

Speech, and especially spontaneous speech, contains many common phrases, such as "Yes I do" or "I know". N-gram language models do not capture the longer common phrases, even though there are enough examples of them to estimate probabilities reliably. Some attempts have been made to treat these common phrases specially, typically as single lexical items. Suhm & Waibel (1994) find common phrases automatically. Word sequences are found which, if treated as

phrases, decrease language model perplexity. The huge space of possible word sequences is reduced to a computationally possible set of candidates using a mutual information criterion. The perplexities of word phrase bigrams (in which the lexicon contains both word phrases and regular words) was found to be lower than word bigrams. Perplexities were normalised to account for the fact that phrases contain more than one word. In dictionary sizes of one or two thousand, using around a hundred word phrases was optimal.

- ***Sub-sentence modelling***

Meteer & Iyer (1996) observe that the information distribution in sentences (from the Switchboard corpus) is non-uniform. They point out that sentences often have a given/new or topic/comment structure, and that given information tends to occur in the beginning of a sentence (establishing the topic), whilst new information tends to occur at the end (commenting on the topic). Their method involves finding the dividing point in each sentence. The dividing, or pivot, point is defined as the first strong verb, or last weak one. Some sentences do not have a verb, and these are divided into two categories: complete (for example, "Yeah" and "OK") and incomplete. Distinct language models are used for each category: before pivot, after pivot, no pivot (complete) and no pivot (incomplete). The distribution of types is very uneven - in their data there are around 300 000 words in each of the before and after pivot categories but only 10 000 words in the no pivot (incomplete) one.

Meteer & Iyer then build a finite state model of conversations in terms of these four categories. The model is a simple loop. Refinements are made to account for sentences which begin turns, and for sentence connecting elements. This model only uses *within* sentence constraints since sentences are not themselves categorised; it has bigram transition probabilities, which is somewhat simpler than the best dialogue model chosen on page 124.

4.6.2 Decimating the training data

Unfortunately, using sub-language models based on move types means dividing the training data between the models. So the advantages of better language modelling are potentially offset by lack of training material. However, the strategies for compensating for lack of data described in section 4.3.1 can be applied to alleviate this problem. I will show in chapter 7 that the trade-off is in favour of utterance type-specific language models.

4.6.3 Use

How do we use a language model composed of a set of sub-language models? For our telephone number grammar example, the sub-grammar is simply inserted into the main grammar, expanding the "telephone number" item in terms of words. The same thing can be done here, but now our main grammar simply contains one item for each move type, and each item has its own sub-language model. This is made clearer in figure 4.3. The transition probabilities between the models are given by a dialogue model (see section 6.2) – in the example shown this is just a bigram model.

One thing remains: how do we compute the probabilities in the main grammar? These are the probabilities of the various move types following each other – a dialogue model, as described in chapter 6. Of course, figure 4.3 shows only a bigram language model but, since any N-gram model can be represented as a finite state network (see page 51), the idea is the same for an N-gram dialogue model.

4.6.4 Language model estimation

The *type* of utterance is incorporated into the language model using the utterance type classification scheme introduced in chapter 2. Before conducting lengthy

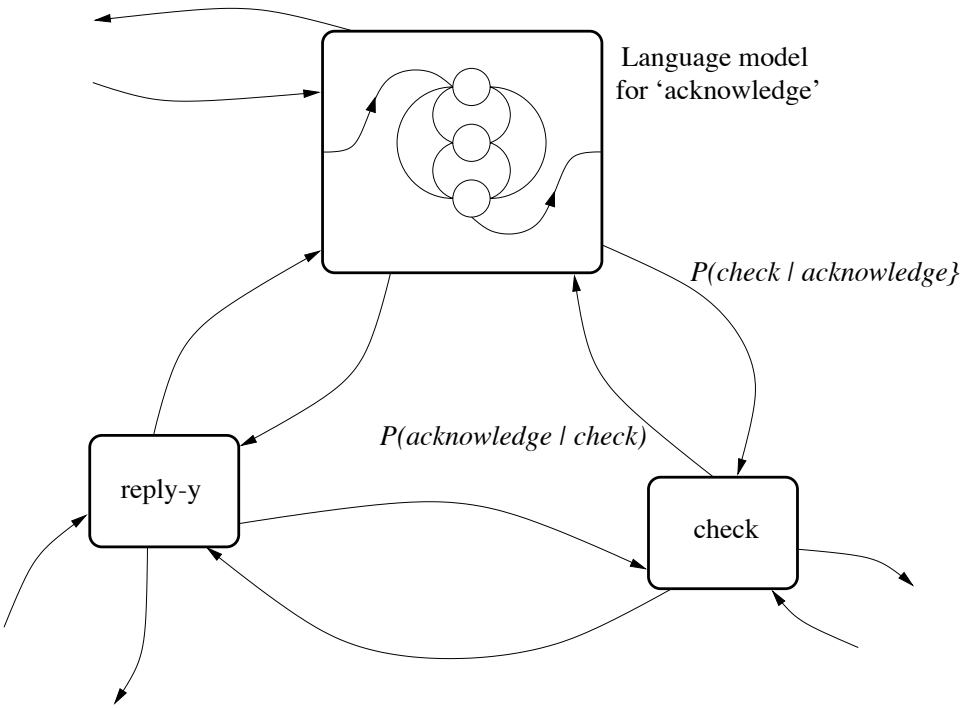


Figure 4.3: How sub-language models form a single model (only some models are shown)

recognition experiments, it is necessary to show that there is likely to be an improvement in word accuracy. We do this by examining language model perplexity, and show that a reduction in perplexity compared to the baseline model can be achieved. The baseline language model described in section 3.3.2 has a perplexity on the test set of 23.6. Perplexity and word error rate are correlated, although the exact relationship is unpredictable.

Each utterance[9] in the corpus has been labelled with one of 12 move types introduced on page 22. The corpus is thus divided into 12 smaller sections. The amount of training data available in each is shown in table 4.3 on page 76.

[9]Each utterance contains exactly one *move*.

move type	sentences	words
acknowledge	2607	6363
align	319	1753
check	598	4359
clarify	246	2149
explain	733	6521
instruct	1407	17991
query-w	262	1863
query-yn	703	5748
ready	784	1574
reply-n	262	770
reply-w	331	2937
reply-y	1020	2824
total	9272	54852

Table 4.3: Move type-specific LM training set sizes

- ***Smoothing the models***

Because of the sparse training data, some of the utterance type-specific LMs will be poorly estimated. To alleviate this problem, we can interpolate the probabilities of the type-specific models with probabilities from a more reliably estimated model : the general model. Section 4.3.2 introduced the estimation–maximisation (EM) method for computing interpolation weights as implemented in (Rosenfeld & Clarkson, 1997).

To compute the EM interpolation weights, a test data set is required – the held-out method introduced on page 49 is used. We will need a further held-out set for selecting between models as described later, so some of the training set is set aside for that before splitting the remainder into training/testing portions.

- *The interpolation weights*

Table 4.4 shows the weights[10] used in smoothing the LMs. Weights close to 1 mean

[10]Averaged over the three held-out combinations described earlier, and rounded to one significant figure because the three estimates typically exhibit variation. Rosenfeld (Rosenfeld,

move type	weight
acknowledge	0.8
align	0.5
check	0.4
clarify	0.3
explain	0.5
instruct	0.7
query-w	0.6
query-yn	0.6
ready	0.9
reply-n	0.9
reply-w	0.4
reply-y	0.8

Table 4.4: The interpolation weights

that the smoothed model will be closer to the unsmoothed move type-specific LM than the general model. The practical method I have used for combining two language models is to weight the training data. The models are estimated by counting the N-grams occurring in training data. These counts can be combined by a weighted sum thus (a, b is an observed bigram, $C(\cdot)$ is a counting function[11] and w is the weight):

$$C_{\text{combined model}}(a,b) \;=\; w.C_{\text{type-specific data}}(a,b) \;+\; (1-w).C_{\text{all data}}(a,b)$$

The combined model is then estimated from the counts $C_{\text{combined model}}(\cdot)$. The size of the weight w reflects two things: how well trained the unsmoothed model is for that move type, and how well the general purpose model models that move type. If moves of a particular type have a very specific grammar, and the LM for that type is well trained, we would expect a weight close to 1; this is the case for "acknowledge", "reply-n" and "reply-y" in particular. Conversely, for move types like "clarify", whose grammar is less specific to that move type (and

1994, page 25) observes that the weights need not be very accurately specified, and suggests 5% accuracy is good enough. I have used 10%.

[11]The counts are normalised for training corpus size

more like a "general" grammar), and/or which have insufficient training data (see table 4.3), the weight will be closer to 0.

Examples of some move types have grammars which appear, on casual inspection of the data, to be similar to the general purpose grammar. If the LMs for these types are well trained, their weights will be close to 1 – instruct, for example – indicating that they do indeed have a particular grammar, if not as obviously as "reply-y", say.

- ***Perplexity***

The individual perplexities of the move type-specific language models are shown in table 4.5. The figures shown are for the *test* set, for consistency with other results shown here. As mentioned above, the test set was not used to choose the best language model.

Test subset move type	**Perplexity**		
	Language model used		
	general	type-specific	smoothed
acknowledge	4.3 *(3.3)*	3.5 *(2.0)*	**3.4** ***(2.0)***
align	22.1 *(19.9)*	31.7 *(24.3)*	**22.1** ***(19.8)***
check	32.4 *(21.2)*	35.5 *(27.5)*	**32.3** ***(21.1)***
clarify	**46.8** ***(29.0)***	60.6 *(46.2)*	46.8 *(29.0)*
explain	**40.7** ***(31.0)***	42.4 *(32.4)*	41.3 *(31.3)*
instruct	41.4 *(29.4)*	**37.2** ***(27.1)***	37.7 *(27.6)*
query-w	36.6 *(26.4)*	34.6 *(24.3)*	**32.3** ***(22.2)***
query-yn	20.5 *(16.6)*	19.3 *(15.6)*	**19.0** ***(15.3)***
ready	4.0 *(3.5)*	**2.6** ***(2.5)***	3.0 *(3.0)*
reply-n	7.5 *(6.1)*	**3.0** ***(3.1)***	3.8 *(3.8)*
reply-w	**24.0** ***(24.4)***	32.3 *(34.6)*	25.0 *(25.4)*
reply-y	7.0 *(4.9)*	**4.6** ***(3.8)***	5.1 *(4.1)*

Italic figures in parentheses are the mean figures for held-out experiments – that is, the figures actually used to select amongst the models. These figures are for models using a set 5 vocabulary and are computed from the mean entropy over 3 held out portions of the training set. ***Bold*** *figures indicate the models selected.*

Table 4.5: Perplexity of general and move-specific models over the test set.

In table 4.5, we see that the perplexities vary widely between move types. The move type-specific language model perplexities are sometimes much higher (worse) than those for the general model. This is the case for "align","clarify" and "reply-w" in particular. I presume this is because of insufficient training data for these types. Furthermore, the smoothed language models do not always have lower perplexity than the corresponding unsmoothed ones. The EM method described in section 4.3.2 only guarantees not to increase perplexity on the *training* set.

We can now select one of three language models for each move type: the general model, the unsmoothed move type-specific model or the smoothed model. The choice is based on perplexity on a held-out portion of the training data, and the general model was chosen for "clarify", "explain" and "reply-w", the unsmoothed models for "instruct", "ready", "reply-n" and "reply-y", and the smoothed models for the other move types. We call the resulting model composed of these move type-specific models the *"best choice"* model. The figures in table 4.5 would suggest the same decision, based on the test set.

- ***Results***

Model	**test set perplexity**
general (baseline)	23.6
original move type-specific	22.1
smoothed move type-specific	21.5
best choice move type-specific	21.0

Table 4.6: Language model perplexities

By combining the perplexities of the move type-specific LMs (by computing the overall mean entropy on test data), we can estimate the perplexity of the new language model which consists of move type-specific sub-models.

Table 4.6 shows those results. The baseline system is the general purpose LM; the original move type-specific model consists of the unsmoothed move type-specific LMs; the smoothed move type-specific LM uses *only* interpolated LMs;

Language model	Log probability
acknowledge	-1026.4
clarify	-996.8
instruct	**-906.9**
query-yn	-989.0
reply-n	-1863.7
reply-y	-1574.8
align	-987.0
check	-1019.8
explain	-996.8
query-w	-992.8
ready	-1750.5
reply-w	-996.8
general	-996.8

Correct transcription: *"Go approximately one inch to the left of the telephone booth." Correct move type is* ***instruct.***

Figure 4.4: Language model component of recogniser output log probability for an example utterance according to various language models

the *"best choice"* model is as described above. Again, perplexities are quoted for the test set.

There has been a sufficient perplexity reduction (especially for the best choice model) to expect improved word accuracy speech recognition. Of course, the perplexities above assume 100% move type classification, and this will not be the case in a fully automatic system where we will choose the LM corresponding to the *recognised* move type.

- *Parameters*

In the 12 backed-off move type-specific language models there are a total of around 50 000 parameters[12], compared to around 10 000 in the general model (page 42), and a total of 9.7 million (12×900^2) bigram probabilities.

[12] 41k bigrams plus 9k unigrams

- *Example*

To illustrate the move type classification power of the move type-specific language models, figure 4.4 shows an utterance and its log probability according to the various language models which make up the best choice model, plus the general model. Note the the log probabilities shown are the language model component of the recogniser output, in other words, they are for slightly differing word sequences – see table 7.3 on page 145 for the corresponding word sequences. The example utterance in table 4.4 is of type *instruct* and it can be seen (in bold) that the instruct language model assigns the highest probability to the utterance.

- *The chosen model*

The *"best choice"* model from above has the lowest perplexity on held-out data – in particular, its perplexity was lower then that of the *"general"* model – so it was chosen as the model for the integrated system. Experiments using this model in the system are described in chapter 7.

Chapter 5

Intonation

5.1 Introduction

In section 2.1 I introduced the concept of categorising utterances by *type* as a way of using constraints at and across the utterance level in speech recognition. These constraints could include intonation, and I proposed that one effective way to use intonation was as an indicator of utterance type. Durational information (whether of segments or phrases) could also be used, particularly in relation to boundary detection. Here I will briefly review uses of both intonation and duration.

The use of intonational information to aid speech recognition assumes some relationship between intonation and syntax or semantics. I describe some aspects of this relationship, and some of the attempts to use intonation in relation to structure (syntax) and content (semantics). First, some examples of these relationships are given, and some of the frameworks for describing intonation are outlined.

- *A note regarding the work in this chapter*

The work using Taylor and Wright's intonation recognition system (Wright & Taylor, 1997), was carried out by Taylor and Wright and not by myself - the

division of labour was described in the introduction on page 9. Furthermore, much of the other content of this chapter is adapted from Taylor, King, Isard and Wright (1998,pending). The intonation recognition provides input to the integrated system, and contributes to utterance type classification.

5.2 Review

5.2.1 Frameworks

Some sort of framework for *describing* intonation is required before we can attempt to *model* it. Normally, accounts of intonation attribute meaning to either entire contours (Sag & Liberman, 1975; O'Connor & Arnold, 1961; O'Connor & Arnold, 1973) or to types of accent, sometimes with rules for composing intonational meaning from accent combinations, for example (Pierrehumbert & Hirschberg, 1990).

For automatic processing, accent detection and contour classification can be done in two stages. This means that the intermediate description is in terms of accents and boundaries. I will concentrate on schemes which compose intonation contours from such elements. The differences between systems are either in the categorisation systems for these accents and boundaries, or in the model for composing intonation contours using them, or both.

Duration effects can be at the segmental level (often referred to as microprosody) or at syllable or phrase levels. Only a brief review of the use of duration is given here, since the system of Taylor and Wright discards most timing information (such as the time between accents, although accent duration itself is used).

- ***Accent and boundary descriptions***

At a local level, description of pitch accents and boundary tones can be either symbolic or parametric. That is, there can be a finite set of possible accent shapes, or a continuously variable space.

- *ToBI*

ToBI (Silverman *et al.*, 1992; Beckman & Ayers, 1994) is a symbolic description scheme with a fixed set of accent and boundary tone labels. Each label describes a different shape of accent. The distribution of ToBI symbols in real data is very uneven, and this means that, despite the number of possible accent labels, the ToBI scheme does not result in particularly descriptive labellings. The following is taken from (Taylor *et al.*, 1998,pending):

> In a study on ToBI labelling (Pitrelli *et al.*, 1994), labellers agreed on pitch accent presence or absence 80% of the time, while agreement on the category of the accent was just 64% and this figure was only achieved by first collapsing some of the main categories (e.g. H* with L+H*). Second, the distribution of pitch accent types is often extremely uneven. In a portion of the Boston Radio news corpus which has been labelled with ToBI, 79% of the accents are of type H*, 15% are L*+H and other classes are spread over the remaining 6%. From an information theoretic point of view, such a classification isn't very useful because virtually everything belongs to one class, and therefore very little information is given by accent identity. Furthermore, not all H* accents have the same linguistic function, and so there are intonational distinctions that are missed by only using a single broad category. Finally, recognition systems which have attempted to automatically label intonation usually do much better at the ac-

cent detection task than at classifying the accents (Ross & Ostendorf (1995), for example).

- *Rise-fall-connection analysis and tilt parameterisation*

In contrast to ToBI, Taylor's rise-fall-connection (RFC) analysis of intonation contours (Taylor, 1992; Taylor, 1993; Taylor, 1994) is parametric. RFC analysis begins by locating rises and falls in a smoothed F_0 contour. Piecewise quadratic curves are fitted to each rise or fall. These curves can be described by their rise and fall durations and rise and fall amplitudes. These are the RFC parameters. From the RFC parameters, Taylor's *tilt* parameters are computed for each intonational event, these are: tilt; F_0 amplitude; duration and start F_0. Tilt is a quantitative description of the accent shape, and ranges continuously from 1 for a pure rise, through 0 for a rise-fall to -1 for a pure fall. The events are located in a separate procedure which is independent of the tilt analysis. This can be done with neural networks or HMMs, for example. For the purposes of event detection, there is only one category of accent and one boundary tone. Additional symbols are used for cooccurring accent and boundary tone, and connecting elements. This system is discussed further in section 5.3.

- *Targets*

Accent and boundary description schemes are implicitly idealised – the canonical forms represented by the symbols are never perfectly realised. In a *target* scheme, this idealisation is made explicit: F_0 is described by targets which it approaches but never achieves.

Campbell (1994) compares Taylor's RFC model with a target-based model; both are used in combination with a segmental duration model. He concludes that neither model (of F_0) contributes much to automatic annotation of ToBI

labels, and that simple F_0 features, such as the local average F_0 and the F_0 change across syllables, can do just as well.[1] However, Campbell's results are reported as percentage of correctly detected accents, and although this figure is higher using only raw F_0 features rather than one of the models, the number of false accent detections is also higher. Campbell notes that the more sophisticated models tend to use too much detail – for example, too many accent labels are assigned by the RFC analysis. This reduces the chance of generalisation from training to testing data, since there are too many unique training label sequences.

- ***Contours composed of intonational events***

All of the theories considered here are of the same type: they describe intonation as an idealised sequence of *events*. The events are typically pitch accents and boundary tones, although there is no real restriction on the type of events that can be generated by such finite state models.

None of the models proposed is directly useful in recognition of intonation because they are not stochastic and therefore have no parameters which can be trained from data. However, a simple extension, adding probabilities of transitions and observations, results in a Markov model which *is* trainable from data. Extending the description to a Markov model does not impose any further restrictions of the type of the events. If each state in the model is then allowed to generate *any* of the symbols, we get a Hidden Markov Model (HMM) in which there are many ways of generating a given observation sequence (accent sequence).

Figure 5.1a shows Pierrehumbert's intonational grammar (Pierrehumbert, 1980). This can be rewritten as figure 5.1b in which symbols (intonational events) are emitted from states rather than arcs. A self transition on the second state allows more than one pitch accent to be generated. Figure 5.1c shows Ladd's (1996)

[1]The Tilt and ToBI systems are quite different, so it is perhaps not surprising that tilt parameters are no better for automatic ToBI labelling than other parameterisations.

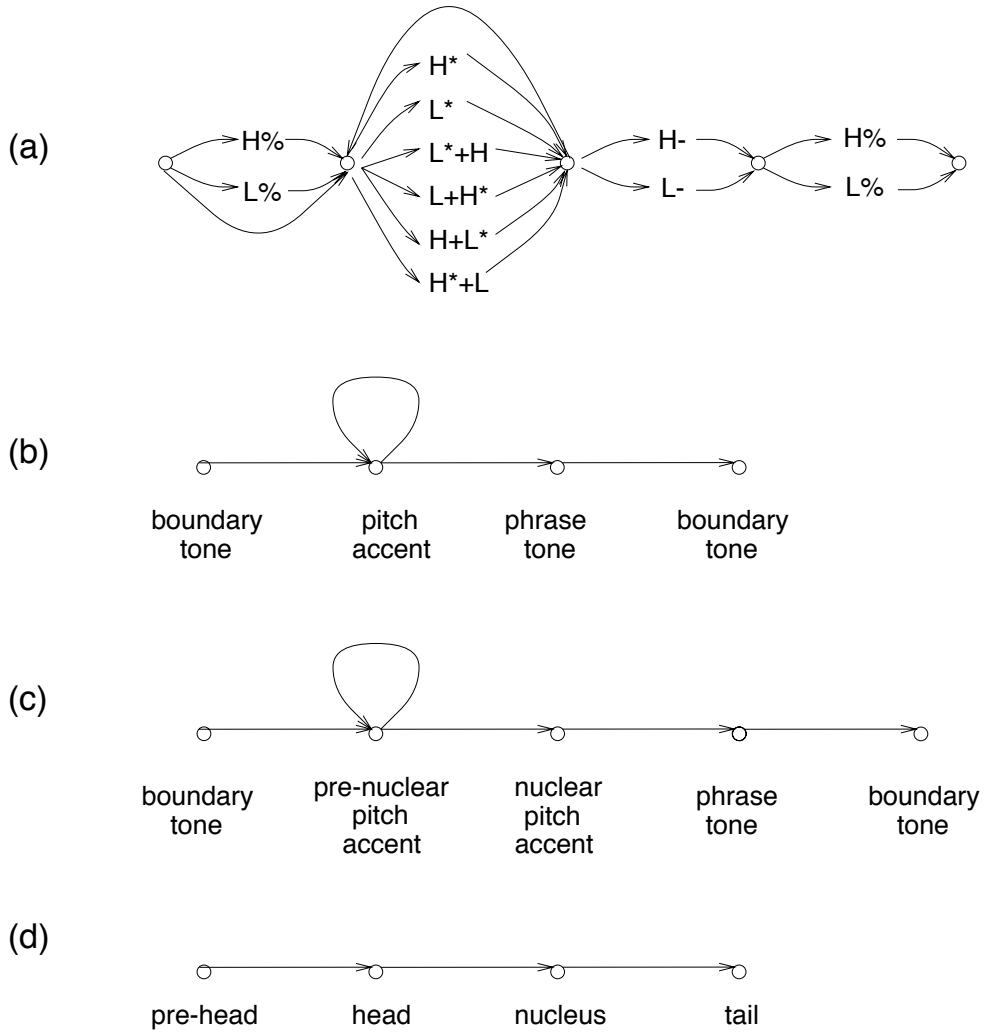


Figure 5.1: Finite state models of intonation structure, from (Taylor *et al.*, 1998,pending)

amended version of Pierrehumbert's model. The traditional description of British English intonation contours is shown in figure 5.1d.

- *Modelling the British School scheme*

The F_0 contour in each segment is described as one of a finite set of patterns; there are 15 patterns in the 1961 O'Connor & Arnold scheme (1961). Jensen *et al.* (1993) modelled the stylised patterns of O'Connor and Arnold using HMMs,

omitting the four symbols for minor events. The data used was a small set of sentences recorded by a single speaker[2]. In a transcription task, 67% accuracy was achieved (accuracy accounts for false insertions) which is a usable recognition rate, but speaker dependent.

- ***Global contour descriptions***

Descriptions of intonation can be of entire contours, or intonational *tunes* as in (Sag & Liberman, 1975), where the tunes are used to disambiguate speech acts.

- ***Syllable level***

Isolated pitch contour patterns have been modelled using HMMs (Ljolje & Fallside, 1987b; Ljolje & Fallside, 1987a). Monosyllabic words were recorded with one of four simple pitch patterns: rise, fall, rise–fall, fall–rise. HMMs were then trained to recognise these patterns using four parameters: F_0, Energy and their first derivatives. Accuracy was 93%, but these patterns were very simple, and isolated (so the task was classification, not recognition). Associating intonation patterns with syllables means that, for automatic intonation recognition, a segmentation of the signal is required, as in (Strom *et al.*, 1997), for example.

- ***Segmental level***

At the segmental level, the use of intonation is limited because intonational events are generally associated with whole syllables or longer units. However, F_0 clearly has some consistent effects on segment properties which can be exploited.

On the other hand, duration information can be extracted at the segment level. The durations of segments within a syllable are governed by some rules, see (Campbell & Isard, 1991) for example, and syllable durations and timing reflect

[2]This is not realistic data, because the speaker could produce the desired contours on demand.

global utterance properties such as speech rate. In short, prosodic effects at the utterance level are reflected in segment durations.

- *Pitch*

Bartkova (1997) uses both segment duration and pitch movement in a word spotting system to evaluate the likelihood of recogniser output. Keywords are likely to occur in a stressed position at a prosodic boundary, and therefore F_0 information can be used to assess the hypotheses output by a keyword spotter. Unfortunately, the word spotting application is not an ideal candidate for this technique since only isolated words are available to the prosodic rescoring algorithm (the remaining speech having been classified as non-keyword by the recogniser and therefore discarded).

- *Energy*

Accents can be detected without using F_0. Kondo (1995) uses accent dependent models for connected digit recognition. The grammar is modified to constrain allowable sequences of accented/non-accented digits and improvements in recognition accuracy were obtained. F_0 was not used as a parameter in this work, with accents being distinguished using energy alone.

5.2.2 Relation to structure

Intonation gives cues to the structure of utterances, and their relationship to one another. That is, intonation carries information about both the content of the current utterance, and its rôle in, say, a dialogue.

- ***Syntax***

Intonation and prosody clearly relate to syntax, since in spoken language all cues to structure must be signalled acoustically. Furthermore, since spoken language is rarely "grammatical", these cues to structure are crucial to the listener for disambiguation and rapid processing.

  - *Word boundaries*

In spontaneous speech, where word boundaries are generally not marked with pauses, there are durational and intonational cues instead. Muteanu *et al.* (1997) use F_0, energy and duration of pitch events to recognise and classify such boundaries. In this system, high reliability is preferred over high detection rate, and around half of the boundaries are correctly identified. These boundaries are intended as anchor points for speech recognition and higher level processing – a similar idea to the division of speech into utterances.

  - *Syntactic boundaries*

To date, the use of duration and intonation in connection with syntax has been to select between alternative parses of a single word string rather than between alternative word string hypotheses. Strangert (1997) shows that prosody is used to signal boundaries. Dogil *et al.* (1997) discuss the relationship between prosody and discourse structure, concluding that those intonational boundaries which are significant in the discourse are more strongly marked than those which are not. This leads to the possibility of segmenting discourse into smaller sections, as discussed in Warnke *et al.* (1997); this was considered in more depth in section 1.2.2.

So, pause information is useful in speech recognition for breaking the input into smaller chunks for faster processing, better language modelling (see chapter

4) and integration of longer term constraints (see section 2.1). These chunks bear some relation to syntactic units, which may be helpful in later parsing or language modelling. Takagi & Itahashi (1995) divide the input spontaneous speech into utterances using silence detection, and then use language models for utterances, with "pause" as a pseudo-word. The recognition rate (quoted only for content words) increases using these two techniques.

We have seen that spontaneous speech has natural breaks, allowing division into smaller units (the definition of these units was considered in section 4.6.1). These units are a convenient size for *both* modelling and processing; reliable identification of pauses provides anchor points for speech decoding.

- *Analysis by synthesis models*

A significant problem in training models of prosody, and indeed of syntax, is the generation of sufficient training data. Typically, hand labelled data is used, but this is expensive. Hunt (1993; 1996) presents prosody-syntax models which can be trained without the need for prosodically labelled data, although correct syntactic trees are required for each training utterance.

Hunt's prosody-syntax models derive from those of Veilleux *et al* (Veilleux & Ostendorf, 1992; Ostendorf *et al.*, 1993) which have the form of two prosodic label generators (decision trees, for example). One generates (a probabilistic distribution of) prosodic labels from acoustic features, and the other from syntax. A comparison of the two prosodic descriptions indicates the likelihood that the syntactic parse used was correct. The prosodic description system is based on a discrete set of break indices. Data labelled using this system is required to train the two generators.

Hunt's models are different in that they use a scalar prosodic representation for the comparison. Two versions of this description are generated, one from

syntax, the other from the acoustic signal, and they are compared as before. Hunt demonstrates that models can be trained without explicit prosodic labels, and the scalar intermediate representation is then learned by the model. In other words, rather than use prosodically labelled data to specify the intermediate description, this description can be learned by the model itself

A high correlation is shown between the syntactic representation and low level acoustic features and thus high accuracy in resolving syntactic ambiguity is achieved using those acoustic features. The acoustic features used include numbers of syllables in words, pause durations, and features of the pre-boundary syllable such as number of phonemes, nucleus duration, energy measures and so on. These all require a phonetic transcription, and for training this is obtained by forced alignment recognition.

- ***Dialogue***

Intonational cues to dialogue structure are part of a mechanism for controlling conversational interaction between two people. Speakers can indicate regions where the listener should pay attention, perhaps because new information is being given; intonational cues are given to indicate phrasing or breaks. In a dialogue situation, these cues may signal that the speaker either wishes to continue, or wishes to hand over to the other person. The perceived relationship between intonational markers (pitch accents and boundary tones) and information structure has been shown, for example by van Donzel & Koopmans-van Beinum (1997).

Buder & Eriksson (1997) claim that the the rhythms of conversational speech are carried *across* turn boundaries. That is, the two speakers somehow "fit" their speech together; they also claim that the effect is present in all languages. The timing of these rhythms are controlled in part by the need to breathe (breath phrases) and partly by some social constraints.

The relationship between intonation and dialogue structure is also of interest

in speech synthesis (Black & Campbell, 1995; Bruce *et al.*, 1995; Hirschberg *et al.*, 1995).

5.2.3 Relation to content

- ***Word level***

The information content of words is reflected in their intonation. In particular, *focus* is often marked by pitch events – (Elsner, 1997), for example. Typically, new information is more intonationally marked, and more clearly spoken, than already given information.

- ***Utterance level***

This is the area of focus for our work: the use of intonation information at the utterance level. Utterances can be classified by content and function, as discussed in section 2.2. Without knowing the words, intonation is a major cue to utterance type.

The signalling of utterance type by both global properties of F_0 and local terminal pitch rises is examined in (van Heuven *et al.*, 1997). Examples of statements and 3 types of question (yes-no, wh- and declarative) were automatically classified. The yes-no and declarative questions were found to be the two most confusible types.

- *Dialogue rôle*

Kowtko (1996) has shown that intonation carries some, if not sufficient, cues, to the *type* (as defined on page 22) of Map Task utterances. Kowtko's thesis attempted to describe the function of intonation in task oriented dialogue. The foundation of this work was the theory of Conversational Games (Kowtko *et al.*, 1993). The findings from (Kowtko, 1996) lead us to believe that there are sufficient

acoustic cues to utterance type to allow automatic classification of utterances into types using the acoustic signal. Furthermore, it was also clear from Kowtko's work that the relationship between intonation and utterance type is not a straightforward mapping, and that dialogue context plays an important rôle. In other words, an utterance's rôle in the dialogue can be signalled by intonation, but the intonational pattern used depends on the dialogue context. Since F_0 is straightforward to extract[3], and process into a usable form (as shown in chapter 5), intonation seems to be an ideal candidate cue to utterance type.

A more comprehensive modelling of pitch events has been performed by Wright and Taylor (1997) where hidden Markov models (HMMs) are used to model sequences of pitch accents and boundary tones. A HMM is constructed for each of 12 types of utterance and trained on spontaneous speech data from a dialogue context. Classification accuracies of over 60% are reported when used in conjunction with an N-gram utterance type sequence model (around 40% accuracy for isolated utterances). This model is discussed further in section 5.3.

5.2.4 Using prosody and intonation for speech recognition

As already mentioned, prosody and intonation provide extra information which we can utilise in speech recognition. But what *do* prosody and intonation do for speech recognition – for human listeners as well as machines? Hess *et al.* (1996) provide a useful review, in which they claim that they do two things for us: they disambiguate and constrain. In the speech recognition–as–search paradigm I introduced earlier, these amount to the same thing. In automatic recognition systems, the sheer number of sentence possibilities is a major problem. Additional constraints reduce the size of the search space leading to faster and better solutions. At many levels, from the segmental to semantic, intonation and prosody distinguish between ambiguities.

[3]We are using high quality and relatively clean speech.

- Segmental level: segment duration is a distinguishing feature
- Word level: lexical stress differentiates, for example, the noun and verb forms of "permit", "construct" and "segment".
- Phrase level: intonation puts in the "punctuation marks" which provides syntactic constraints
- Sentence level: modality is often intonationally marked. Modality constrains syntax
- Discourse level: Topic changes and new words are signalled by intonational marking – for example (Wichmann *et al.*, 1997).
- higher levels: for example, the speaker's mood.

Prosodic and intonational information can be used in the speech recognition process in one of two ways: two passes or an integrated method. Typical two-pass approaches are to use prosody and/or intonation to segment the speech or mark prosodic phrase boundaries prior to recognition, and to use intonation in a rescoring pass after speech recognition. In an integrated approach, F_0 and speech recognition are used simultaneously to obtain some combination of word sequence, syntactic or prosodic phrasing and intonational description.

- ***Two-pass approaches***

Hirose *et al.* (1994) propose two schemes for using F_0 information to improve speech recognition. Their first method is to find syntactic boundaries using only F_0 and energy, although in this they only go as far as detecting the boundaries and do not apply this to actual word error rate reduction. The method is basically a rule-based analysis of macro- and microscopic features of F_0. The pitch contour is segmented at dips (minima) in the energy contour, and rules are applied to

generate candidate syntactic boundaries at dips in F_0. A number of thresholds are used in the rules, and these can be adjusted to control insertion errors (a problem typical of automatic accent or boundary detection). The system detects just over 80% of syntactic boundaries with an insertion rate of 30%. The second method in (Hirose *et al.*, 1994) involves analysis by synthesis to choose amongst sentence hypotheses. Candidate word string hypotheses are used to generate F_0 contours. Contours are only produced for the parts of hypotheses where there is ambiguity (about the words). The sentence hypothesis whose synthesised F_0 contour most closely matches the real contour is selected. In (Hirose & Sakurai, 1996), the F_0 contour is smoothed to eliminate microprosodic effects, and speaker adaptation of the F_0 generating model is used.

A common observation in speech recognition is that longer utterances are more likely to be incorrectly transcribed, and they also require more CPU time and memory to process. Vereeken *et al.* (1997) propose that chopping utterances into smaller *prosodic phrases* prior to recognition would therefore increase the subsequent accuracy. The basis for finding these prosodic phrases is based on a syllabification of the speech followed by finding silences longer than 150ms, breaths, clicks and other background noises. Small reductions in phonetic recognition error rate are reported.

- ***Integrated approaches***

The interface between the speech recognition component and "higher" components such as the parser, semantic interpreter and so on, is quite important. As in all fields of pattern recognition, such interfaces started off as what I will call "hard decision" and have moved to "soft" or probabilistic ones. For speech recognition this means moving away from simply passing the most likely word sequence out of the recogniser, to giving either an ordered (and possibly scored) list of alternatives, or a *lattice* – see figure 5.2 on page 97 – which efficiently represents various

alternatives. So-called "phonetic typewriters" attempt to recognise the phone sequence, which can then be decoded into a word sequence (Kohonen *et al.*, 1988; Jitsuhiro *et al.*, 1995), but it is now widely accepted that this is not the best approach. The recognition of segments and their decoding into words must be integrated. The main advantage claimed for the phonetic typewriter approach is that the recogniser is "vocabulary free" – that is, there is no lexicon or word string language model. Unfortunately, these are the very components which can significantly improve recognition accuracy.

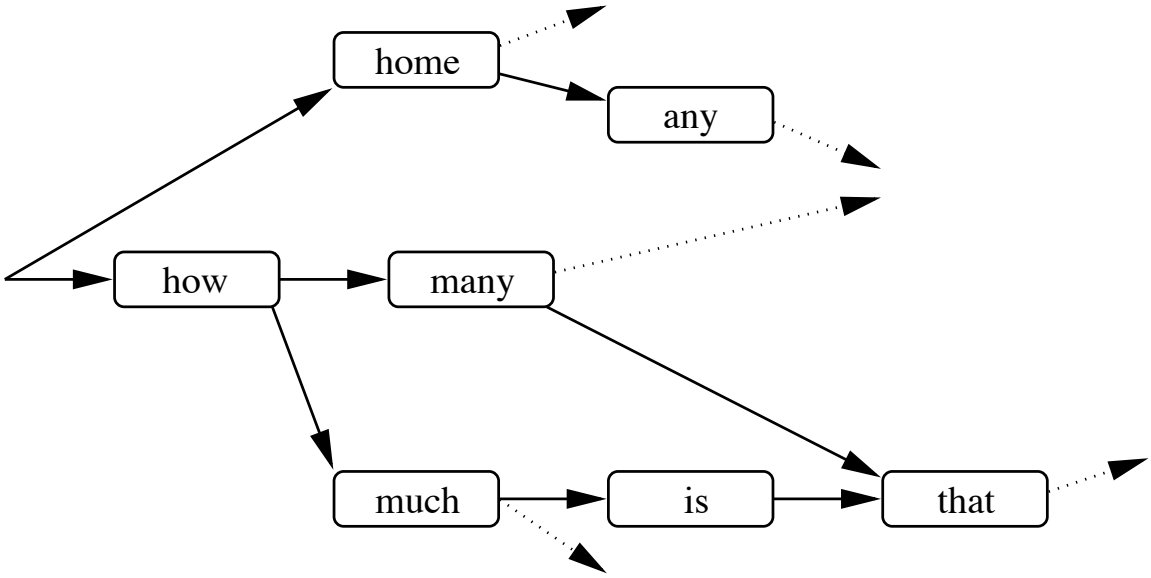


Figure 5.2: A lattice representation of word string hypotheses

Integrating prosodic or intonational information into speech recognition should therefore be in a probabilistic framework, which means that the "score" assigned by the model of prosody or intonation should be combined with the acoustic and linguistic "scores" as a product of probabilities (or, more likely, as a sum of log probabilities), rather than be used as some threshold in a decision rule. Of course, the models will not actually estimate true *probabilities*, but this does not matter; this is explained in the description of the practical solution on page 135.

- *Use of microprosody*

Dumouchel and O'Shaughnessy (1993) describe the use of segmental prosody in a large vocabulary recogniser. The use of F_0, intensity and duration in this system is purely at the segmental level. All other things (such as intonational context) being equal, low vowels tend to have low F_0, and high vowels tend to have higher F_0. Open vowels are more intense (have more energy) than close vowels. These effects are modelled as normal distributions and used to assign probabilities to sentence hypotheses via a (weighted) product of acoustic, language model and prosodic probabilities. This method of integration into a speech recogniser has a solid theoretical (Bayesian) foundation. Only a small improvement was reported (3% increase in recognition rate, although the absolute rate was not given, and it is not stated whether this figure is relative or absolute). Clearly, this use of prosody is very limited; nothing is made of effects spanning more than one segment (a phone or diphone in this case).

5.2.5 Summary

In conclusion, because intonation is such a complex phenomenon, and relates to many things – from segment type to the speaker's mood – the best way to integrate intonation with speech recognition *is* for utterance type identification. This conclusion is supported by the literature, including analysis-synthesis methods such as those in (Campbell, 1994) where the intonation contour is treated as a property of an utterance rather than of individual syllables. This approach has the further advantage that intonation modelling is independent of the segments in the speech, and therefore **does not require speech recognition.** I will show that this approach is effective using the experimental system in chapter 7.

5.3 Automatic intonation recognition

5.3.1 Introduction

This thesis presents a novel method for using intonation and other information for automatic recognition of spontaneous dialogue speech. Speech recognition is treated as a search problem, to find a solution satisfying a set of constraints. These constraints are combined in a weighted probabilistic fashion, and therefore this method requires a probabilistic model of intonation that estimates the probability that an unknown utterance is of each of a set of utterance types.

Here I describe an automatic intonation recognition system developed by Taylor and Wright and described in (Taylor, 1993; Wright & Taylor, 1997). The system estimates the probability of an utterance being of a particular type in an entirely bottom-up fashion from the speech signal. It consists of two distinct processes: in the first, pitch accents and boundary tones are first found and then given parametric descriptions; in the second, these pitch accent and boundary tone (collectively known as intonational events) descriptions are modelled, and probabilities of move types are computed. These two processes are called event labelling and intonational tune modelling.

5.3.2 Event labeller

In section 5.2.1, intonational event description schemes were divided into two categories: symbolic and parametric. Taylor's *tilt* (Taylor, 1998) scheme is one of the latter. The method for event labelling consists of two components: an event detector, and subsequent parameterisation.

- ***Detection***

For the purposes of intonational event detection, there is only one class of pitch accent (a), and one boundary tone (b). Further symbols are required for accent and boundary tone together (ab), connections between them (c) and silence (sil). The F_0 contour is heavily smoothed before accent detection.

Taylor's most recent system (Wright & Taylor, 1997; Taylor *et al.*, 1998,pending) uses Hidden Markov Models (HMMs). One model is trained for each of the symbols above. The state observations are F_0, energy and their first and second derivatives. Assessing the performance of such systems is problematic. Not only must events be reliably detected, insertion of spurious events must be minimised and the events must be in the right places. The system described in (Taylor *et al.*, 1998,pending) was assessed by computing the overlap between detected events and hand labelled ones. 74% of events overlapped by 50% or more. However, if minor events are ignored, this figure is 86%. Unfortunately, there are a significant number of insertions, making the accuracy only 32%. This may not be as bad as the figures suggest, because most inserted events will be minor ones. This will be represented in the subsequent parameterisation. Since the models of utterance type intonation contours are trained on the output of the automatic event detector, these inserted events may not degrade performance much. This is because the rate of minor event insertions is similar for all utterance types, so these events will not be used by the intonational tune models.

- ***Parameterisation***

The event recogniser produces a sequence of labels and their start and end times. Each event so labelled is then parameterised using the Tilt intonation theory (Taylor, 1998). Thus each event is represented by 4 parameters: tilt; F_0 amplitude; duration and start F_0. These parameters form the input to the second process –

they are the observations from the utterance type intonation contour models.

5.3.3 Intonational tune to utterance type

Models of the intonational tune (that is, the event sequence output by the event detector) for each utterance type are used to compute the probability of each utterance being each of the types (Wright & Taylor, 1997). Both training and testing data has been pre-segmented into utterances. These utterance type intonation contour models generate sequences of parameterised events in the same way that models of phonemes for speech recognition generate spectral descriptions of the speech signal. As in speech recognition, HMMs are used. One HMM is trained for each utterance type (for example, the 12 move types of the Map Task). The observation vectors of the HMMs consist of the 4 parameters listed above. Models for all utterance types have 3 states. The precise placement of the events in time is discarded, but their chronological order is retained. Because the space of possible intonational tunes is infinite, the HMMs compute only relative log likelihoods of the utterance types. However, the likelihoods computed by the intonational utterance type classifier will become part of a weighted sum in the log domain in the move type sequence classifier described in chapter 2 anyway, so normalisation is unnecessary. Referring to equation 7.8, the intonational utterance type classifier computes L_i^I. Results of utterance type classification using this system based on intonation alone are given on page 141, and as part of the whole system in table 7.1 on page 141.

Chapter 6

Dialogue

6.1 Introduction

Having decided that automatic classification of utterances into types is the goal, and that dialogue-motivated types should be used to allow dialogue context constraints to be imposed, I now consider dialogues, in theory and practice. The purpose of this chapter is to produce a dialogue model which will be a constraint for automatic classification of utterances into types.

- *Organisation of this chapter*

I will start by outlining some of the situations in which dialogues arise. I then consider modelling spoken dialogues;such models can be used as constraints for utterance type classification, and consequently for speech recognition. A selection of theoretical frameworks from the literature for analysing dialogues are considered, and I summarise Power's theory of conversational games. I then show how part of this theory can be used for statistically modelling dialogues. Then, I describe the database chosen for experimental work (for speech recognition and language modelling, as well as dialogue modelling), and go on to describe the building of dialogue models using this data. Results are given for a selection of models, and the best model is taken forward into chapter 7 where it is used in the system to

improve speech recognition accuracy.

- ***Task***

For automatic recognition of read text, the only applications are dictation or command and control. In the case of dialogue speech, several tasks can be defined, depending on whether the computer is an active participant or passive "overhearer", and what the relationship between the participants is. As we saw in the introduction, goal oriented dialogues are of particular interest, for applications such as information retrieval, co-operative planning (TRAINS, for example), and clarification dialogues in machine translation systems (such as Verbmobil).

- *Giver/follower tasks*

Typically, goal-oriented dialogues involve two participants. One of the participants tends to give instructions, the other to follow them. Even in co-operative, mixed-initiative dialogues, one participant has overall control, or "the last word" we might say. In the TRAINS system, the human operator has ultimate control.

The Map Task (Anderson *et al.*, 1991; Bard *et al.*, 1995) dialogues are mixed-initiative, goal oriented dialogues, but with a marked contrast between the two participants – the *giver* and the *follower*. The instruction giver generally has control of the dialogue, and initiates "sub-dialogues" to achieve sub-goals. On page 112, I will introduce the theory of conversational games which can be used to analyze such dialogues.

In the human-computer interaction scenario, the computer could be either one of the active participants, or a passive "overhearer". If the computer were the giver, the task might be an automatic route planning service in which the user is being instructed in the best route by the system. An application in which the computer is the follower might be the TRAINS system. An example of the

overhearer task is Verbmobil. So, all these variants have real world applications, and since they are closely related, I will address the "overhearer" problem since it is the most general and involves recognition of both giver and follower, therefore including both variants of the participant task. Should the computer be a participant in the dialogue, it will always know with 100% accuracy what it said, and what its intention was; therefore this task is easier than the overhearing one. Also, as mentioned on page 105, there are problems with collecting data for training and testing human-machine systems which are can be avoided by using human-human dialogues.

- *Information transfer tasks*

One of the most common uses for human computer interfaces is the accessing of information, typically from a database stored on, and searched by, the computer. A dialogue is a good way to elicit requirements from users, avoiding the need for them to formulate database queries (typically in a specialised language) themselves. The information transfer then is in fact two way - the computer asking the user questions in order to refine a query, and the user getting information from the database.

An early example of this application was the Resource Management task - a very constrained task involving simple requests for information about naval resources. Although this is really a command and control task rather than a two-way dialogue, the intended application was a hands-free information management system. The vocabulary, and indeed grammar, was fixed and the speech used was read text. This task proved very popular as a benchmark – see section 3.2.2.

A more realistic example of a goal oriented dialogue might be:

> "Do you have any information about Edinburgh ?"
> *"Is that Edinburgh, Scotland ?"*
> "Yes, it is."
> *"I have information about hotels, festivals and the castle."*
> "Tell me more about hotels."
> *"What type of hotels ?"*
> "What types are there ?"
> *and so on ....*

The query is resolved through an interaction with the computer. Without this interaction, the system would simply have to display all available information in response to every query, which will often mean presenting large amounts of data – as anyone who has ever searched the World Wide Web will know. Refining the query through a goal oriented dialogue is an efficient way to resolve the particular query that the user is making.

- ***Human-human dialogues***

Although the task of recognising both halves of human-human dialogues was chosen for practical reasons, there clearly *are* applications for the recognition of dialogues between two people. One application can be found in Verbmobil (Wahlster, 1993) where the system is required to follow the dialogue between two participants, in order to build a model of the dialogue and provide contextual information for possible translation requests.

Since people are better at speech recognition and understanding than machines, human-to-human speech is generally more difficult for speech recognisers because the language is less restricted, pronunciation is less precise, speech is 'sloppy' and so on. It has been observed that when people talk to machines, their speech is somewhat different from that used between two people; to quote Eckert *et al.* (1995), "real users behave weird"! If we wish to investigate the use of dia-

logue structure and the other constraints outlined in the introduction on page 6, we need to collect data – and find ourselves in something of a Catch 22 situation: we cannot collect realistic human-machine dialogues without an automatic speech recogniser and dialogue system! Of course, this is not quite true, as we could simulate such dialogues using a Wizard of Oz arrangement. It seems easier to use human-human dialogues since there will be *two* channels of spontaneous dialogue speech plus data for dialogue modelling.

- ***Manipulating the dialogue***

The constraints placed on an utterance by the preceding dialogue can be manipulated. That is, utterances can be designed to elicit particular responses. For example, questions can be phrased in such a way as to restrict the possible range of responses, and therefore those responses will be easier to recognise. This even gives an advantage when the response is outside the set of expected ones, since such responses will typically be intonationally marked as "non-default".

  - *Questions and answers*

The most obvious place where the respondent's utterance can be controlled, in type, word content and intonational marking, is when asking questions. Let us assume the machine is asking the question, and a human is responding. For an example, we will use the Map Task: one participant needs to determine the presence of some feature on the other participant's map. Here are three ways of asking the question:

1. "What do you have below the cottage?"
2. "Do you have a meadow below the cottage?"
3. "You've got a meadow below the cottage, haven't you?"

The wh-question in question 1 above will elicit a wider range of responses than the other two forms. Question 2 is a yes/no-question, and therefore restricts the expected range of responses to "Yes" and "No" type answers. 3 gives an alternative yes/no-question which will elicit the same range of responses as 2, but more intonationally differentiated: if the answer is negative, then we would expect strong intonational marking since the default answer (that is, the most predictable one) is affirmative. Here are some corresponding responses which might be expected:

1. "I have a lake there."
2. "No, I don't."
3. "**No**, I **don't**."

and here are some unexpected, but not impossible, ones:

1. "**What** cottage?"
2. "I have a **lake** there."
3. "I have a meadow below the **old mine**."

where **bold** words are intonationally marked. From the above examples, we can see that both the vocabulary and syntax of the response are affected by the type of question. In speech recognition terms, we combine vocabulary (words used, and their frequencies) and syntax into the *language model*. So we can say that the language models for responses to various types of question will be different. This was explored in section 4.6. See section 2.4.3 for more about the effect of the preceding utterance on syntax. Dialogue manipulation in Verbmobil (Wahlster, 1993) is described on page 17.

- *Instructions*

Frequently, instructions are phrased as questions, in order to force a response. This keeps the dialogue "flowing" and allows the instructor to efficiently check that the instruction has been understood, and obeyed. For example, simply giving the instruction:

"Cross the river."

does not require a response, and may lead to confusion about whether the instruction was understood, or who now has initiative in the dialogue. Rephrasing the instruction as:

"Can you cross the river?"

forces a response, such as:

"No, I can't."

Participants in spoken dialogues use these techniques to ensure that they remain "in sync" with the other participant, that initiative handover is clearly signalled and so on. Any model of dialogue must allow for this, and not be too rigid about expecting simple question-answer or instruct-acknowledge sequences. I will now consider modelling dialogues, taking this into account.

6.2 Dialogue modelling

6.2.1 Introduction

As we saw earlier, one situation where speech recognition might be useful is the interaction between human and machine in dialogue form. Therefore, since we know that the speech we are trying to recognise is part of a dialogue, we can bring extra constraints to bear, and as in all speech recognition situations, the more constrained the task, the easier it will be.

In chapter 2, constraints were introduced at the utterance level, in terms of utterance type. Therefore, we now need a *model* of the dialogue in terms of utterance type. At this level, a dialogue can be seen simply as a sequence of utterances of various types. This sequence has both local and global structure. If the utterance type set we choose is the *move* type set of Carletta *et al.* (Carletta *et al.*, 1995), then these local constraints are described by the theory of conversational games (Power, 1979), which I summarise on page 112. The global constraints are imposed by overall dialogue structure because the dialogues are goal oriented and not open ended. However, in the framework of conversational games, the local constraints are stronger than global ones, which is preferable since the limited number of training dialogues available precludes any chance of modelling global structure. The form of dialogue model is basically restricted by the method introduced in chapter 2, and this lack of data. It is, however, worth reviewing dialogue modelling in general. As in the field of natural language modelling, there is a vast array of models in the literature, yet almost none of them lend themselves to use in automatic recognition systems which rely on the Viterbi algorithm, which will be used a solution to equation 7.7 on page 135. The main problem with much of the descriptions of dialogue in the literature is that they are just that – *descriptions* rather than *models*.

When humans listen to and comprehend speech, they use contextual information from many sources. In conversation in particular, the *topic* and *local* context play a vital rôle. For example, the topic affects the expected distribution of word frequencies or, more generally speaking, the *language model*. The local context, such as the type of preceding utterances, affects the expectation of the current utterance type. For example, here are some likely and unlikely pairs of successive utterances (uttered by the same speaker):

likely	Most of the plays are very good. Some of the actors could be better though.
unlikely	Most of the plays are very good. What did you have for dinner ?

The same applies to pairs of utterances from differing speakers:

likely	Most of the plays are very good. *Yes, but some of the actors could be better though.*
unlikely	Most of the plays are very good. *What did you have for dinner ?*

6.2.2 Theoretical frameworks

- ***Template models***

Where the structure of a dialogue is rigid, and known in advance, a simple slot-filling approach can work well. This would be the case when the computer (in the case of human-computer interaction) held the initiative and was simply obtaining information to fill the slots by asking questions. This type of approach is efficient, but very limited in its ability to cope with unexpected events. Further semantic interpretation of the dialogue is made easy as the information gathered from the user is already held in appropriate slots.

- ***Finite state models***

Bennacef *et al.* (1995) model dialogue as a sequence of speech acts using finite state networks (derived from context free grammars). These networks are not probabilistic. The speech acts are identified through semantic processing of the word string output from speech recogniser. The dialogue model provides information on the current dialogue state to a dialogue manager, but this does not allow improvement of the word accuracy of the recogniser itself.

- ***Plan based models***

Simple plan-based models work well for short dialogues. A simple dialogue plan might consist of three phases: a beginning, a middle and an end. These phases of the dialogue might be called **opening**, **negotiation** and **closing**. The phase of the dialogue affects the type of utterances. In the opening phase, for example, we might expect greetings and responses ("Good morning", "Hello"), in the negotiation phase questions and answers ("Are you free on Monday?", "No, but Tuesday is okay"), and in the closing phase, agreement ("See you on Tuesday").

However, in more protracted dialogues, like those in the Map Task, this model is too simple. This is because the goal is more complex than the simple example above. The Map Task goal (of drawing a route on a map) is typically achieved by participants through a series of simpler sub-goals. Simple plan based models could be nested, with one for each sub-goal in the dialogue (in the Map Task, this might be the drawing of one segment of the route). The plan based model can be extended to cover a variety of canonical sub-dialogues, which Power calls *conversational games*. This is the basis of Conversational Game Theory.

- ***Conversational Games***

Power (1979) introduced Conversational Game Theory as a way of describing and analysing (spoken) dialogues. In this theory, the dialogue is treated as a sequence of, possibly nested, games. For example, if A and B are the two participants in the dialogue:

A "Go across the wooden bridge."

B "Okay."

In this case, A is the instruction giver and B the follower. The conversational game theory analysis of the example would be that it it an *Instructing* **game** containing the two **moves** "instruct" followed by "acknowledge". There are 6 types of game (identified by their opening move type) and 12 types of move in the theory – the 12 move types were introduced in section 2.2 on page 21 for utterance type classification. The move types reflect the rôle of utterances in the dialogue.

Here is a slightly more complicated example:

A "Go across the wooden bridge."

B "I don't have a wooden bridge."

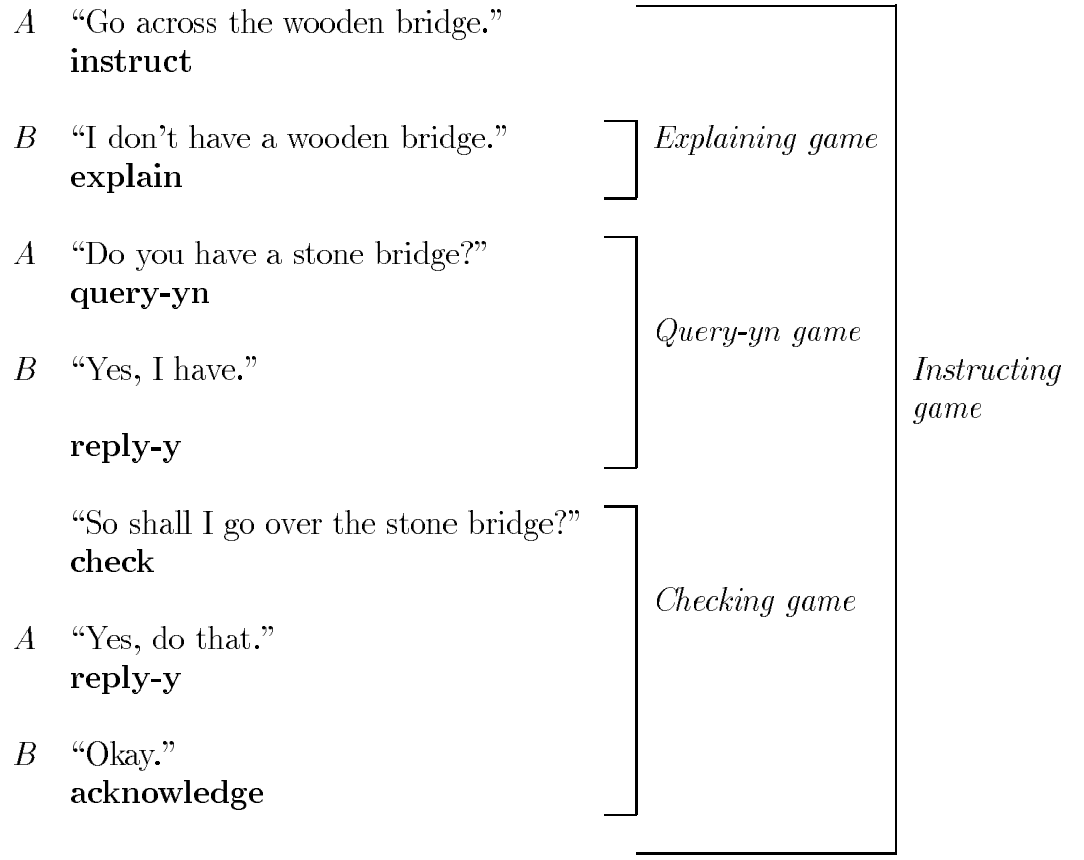
A "Do you have a stone bridge?"

B "Yes, I have. Shall I go over the stone bridge?"

A "Yes, do that."

B "Okay."

In this example, there are nested games inside the *Instructing* game. The conversational game analysis is:



Of course, this theoretical example is much simpler than real-life: speakers do not always take alternating moves; they speak over one another; games (and indeed moves) are not always completed. The theory of conversational games offers two useful ideas: **games** and the **moves.** The 12 move types in (Kowtko *et al.*, 1993), as described on page 21, offer an attractive utterance type classification scheme (discussed on page 21). Modelling games explicitly for use as a dialogue model, and hence as a constraint for speech recognition, has many problems. These are due mainly to the differences between theory and real data listed above, but there is the additional problem that, because games consist of a number of moves, there will not be a large number of example games in the data from which to estimate

a model. This was mentioned in section 6.2.1, in which I described the modelling of dialogues, and the problems associated with doing so from limited amounts of data.

6.2.3 Practical dialogue modelling

- ***Requirements***

Firstly, what are the requirements for the dialogue model? These requirements are due to the method which will be used to apply dialogue context constraints to speech recognition; refer to equation 7.7 on page 135.

- probabilistic
- data-driven – model parameters can be estimated from data
- left context only – preceding dialogue used to predict current state
- finite state representation

The finite state requirement is imposed for the same reason as for language modelling in speech recognition – see page 45. An important consideration in selecting a dialogue model is the amount of data, if any, needed to estimate it. Models which account for global dialogue structure are likely to need a greater number of dialogues for reliable estimation than models which only account for local effects, because (especially in long dialogues, like those in the Map Task) there are many utterances per dialogue – see table 3.2 on page 39.

- ***Models with only local dependency***

The frameworks described so far are *top-down* models of dialogue to varying degrees. They describe the structure of the dialogue. Instead of attempting to model whole dialogues top-down, we can express constraints purely at a local level. Whilst this will not capture the structure of the entire dialogue, it may prove more *useful* and practicable. The definition of "local level" is the key. In section 4.5.2, the problem of segmenting dialogues into useful "chunks" was considered. The utterance definition adopted was the *move* unit of Power's (Power, 1979) theory of conversational games. Although Power's description is top-down (games are composed of moves), the modelling of dialogues as simply *sequences of moves* will capture at least some of the structure of these games. That is, conversational game theory can be approximated by a model of move sequences. This might be compared to the approximation of SCFG grammars by N-gram models (see page 57). In other words, a theoretical model which does not fit the limitations of the algorithm can be approximated by one that does.

- *Game move sequence modelling with N-gram models*

For modelling sequences of moves, there is only really one choice: N-gram models. As described in section 4.3, these models condition probabilities on left context only, and can be represented as finite state networks, thus they are suitable for use in the Viterbi algorithm. In the case of language models, the left context would be the immediately preceding N-1 words; here, it may be any events occurring before the current utterance. I will refer to this left context as the *predictors*. A variety of predictor combinations are evaluated in section 6.3.3.

6.3 The DCIEM dialogues

The task chosen for experimental work was the DCIEM Map Task corpus (Bard *et al.*, 1995), a corpus of of spontaneous task-oriented dialogue speech collected from Canadian speakers of English. This corpus includes some dialogues recorded under conditions of sleep deprivation, but only the "normal" condition dialogues are used for this work.

The DCIEM Map Task (Bard *et al.*, 1995) closely follows the original HCRC Map Task (Anderson *et al.*, 1991). In Map Task dialogues, the two participants have different rôles: one is the instruction giver and the other is the instruction follower. I will refer to these as *giver* (g) and *follower* (f) from now on. The giver has the task of guiding the follower along a route on a map. Each participant has their own copy of this map and the two maps may differ slightly (in the type and position of features) in order to make the dialogues more interesting, as shown in the example pair of maps in figure 6.1 on page 117.

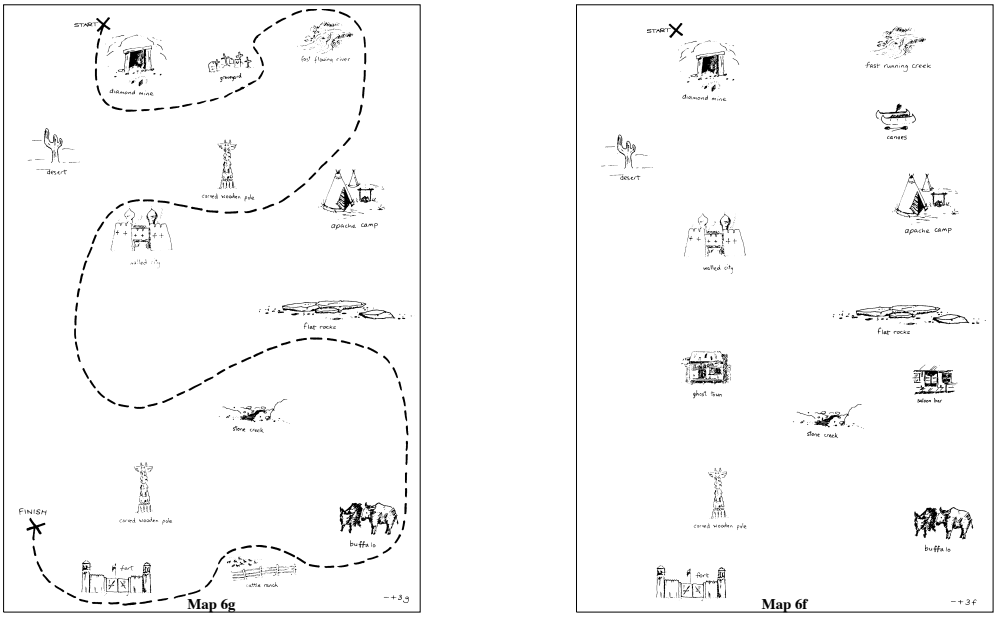
6.3.1 Speech data

The speech in this corpus was recorded using high-quality microphones under controlled conditions (no external noises). Each speaker was recorded on a separate channel, and although there are occasions where the other speaker can be heard, we can treat the two channels as only containing the speech of one of the speakers. The amount of data available is shown in table 3.2 on page 39.

6.3.2 Labelling

- ***Utterance level***

The corpus has been marked up using the theory of conversational games introduced by Power (1979) and adapted for Map Task dialogues as described in



(a) Giver (b) Follower

Figure 6.1: Example maps

(Carletta *et al.*, 1997b) using the set of games and move types defined in (Kowtko *et al.*, 1993). Proposals for the standardisation of such coding schemes can be found in (Carletta *et al.*, 1997a).

For this marking up, the speech has been divided into utterances, each containing a single conversational move. Although there are a certain number of overlapping moves, such problems were ignored in this work, and we treat the division into utterances as reliable and fixed. Automatic methods for division of conversational speech into utterances are becoming available: Warnke *et al.* (1997) for example. The problems of overlapping moves are subject to current research (Bull, 1997).

- ***Word level***

The corpus has word level transcriptions for all dialogues. Since the speech is spontaneous, there are a fair number of disfluencies such as aborted words, restarts and so on, plus non-speech such as coughs and clicks. For our purposes, all non-speech noises were mapped to a special "word" called NW. Some other words (see table 3.1 on page 38) were retained but subsequently ignored when calculating recogniser accuracy. More sophisticated treatment of disfluencies was not possible within the scope of this work – see page 37.

- ***Intonation***

20 of the dialogues have been hand labelled using a simplified scheme which collapses all pitch accents to a single label. The accents are subsequently described in terms of the Tilt scheme, a parametric description of accent shape. F_0 was obtained automatically using the super resolution pitch determination algorithm (Medan *et al.*, 1991) (rather than from a laryngograph). More details about this part of the corpus can be found in chapter 5.

6.3.3 Dialogue modelling

A variety of N-gram dialogue models, as introduced on page 115, was considered. First, "simple" N-grams (all predictors and the predictee are move types) were examined. Figure 6.2 on page 119 shows what is meant by *predictors* and *predictee*. Three methods of using mixed predictors are described, one of which was investigated experimentally.

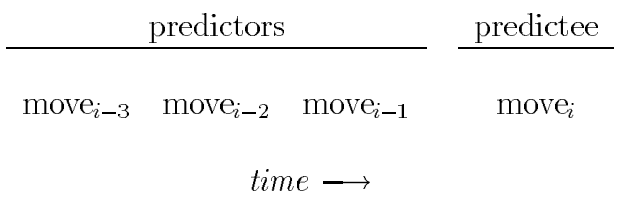


Figure 6.2: Predictors and predictee in N-gram models; move$_i$ is the move currently being recognised.

- ***Simple N-grams***

The vocabulary for the simple models was simply the set of move types. N-gram models were trained for various values of N. Table 4.3 on page 76 shows the number of sentences in the training set – from that table it can be seen that the total number of moves in the training set is around 9200. For a unigram model, the vocabulary size is 12, and for all other N, the vocabulary size is 14 because of the symbols for start and end of utterance. Therefore, for $N = 3$ there are around 2700 N-gram probabilities to estimate, and for $N = 4$ there are over 38 000.

N	Test set perplexity
1	9.1
2	6.3
3	6.1
4	6.8

Table 6.1: Perplexities of simple N-gram dialogue models

Table 6.1 shows the test set perplexity using simple N-gram dialogue models of various orders. For $N > 2$ a floor probability had to be used for unseen N-grams because there were N-grams in the test data which never occurred in the training data. The perplexity of the 4-gram model is greater than that of the 3-gram, indicating that the limit on N has been reached for the amount of training data available.

- ***Mixed predictors***

There is no constraint on the *type* of predictors in an N-gram model – they do not have to be of the same type as the thing being predicted. One available predictor is speaker identity, and this can be incorporated into the N-gram model in 3 ways.

Each item in the N-gram is drawn from a finite set of possibilities: this is the *alphabet*. The alphabet of move types consists of the 12 types in (Kowtko *et al.*, 1993). The alphabet of speaker identities is simply {f, g} – that is, {follower, giver}. Special symbols must be introduced to represent dialogue boundaries (start and finish) – these will be called !ENTER and !EXIT.

- *Method 1 : product of alphabets*

Since each move has one type and one speaker, one way to use both as predictors is to take the product of the two alphabets. This gives 24 symbols, as shown in figure 6.2.

follower_acknowledge	giver_acknowledge
follower_align	giver_align
follower_check	giver_check
etc..	

Table 6.2: Possible alphabet for dialogue N-gram model

To these we must add !ENTER or !EXIT (for predictor and predictee alphabets respectively). However, this method will fail to use current speaker identity as a predictor, since it will be predicting it! For example, a bigram model using this alphabet will contain the bigram (follower_acknowledge, giver_check) in which giver_check is the predictee. The current speaker identity (giver) is part of the *predictee* rather than the *predictor*; it must be moved from predictee to predictor.

speaker identity:	$\cdots$	s_{i-3}	s_{i-2}	s_{i-1}	s_i
move type:	$\cdots$	m_{i-3}	m_{i-2}	m_{i-1}	m_i
					$\uparrow$
time $\longrightarrow$					current move

m_i	type of current move (predictee)
s_i	identity of speaker of current move
m_{i-j}	type of most recent move by other speaker where j is the smallest solution to $s_{i-j} \neq s_j$

Figure 6.3: Notation for heterogeneous N-grams

- *Method 2 : mixed predictors*

An alternative to taking the product of the alphabets is to have a different alphabet for each of the N-1 predictors in the N-gram. The length of the N-gram (i.e. N) can be varied, as can the alphabet for each predictor. There is no requirement for chronological ordering of the predictors. The notation I will use is given in table 6.3. s is speaker identity (follower or giver) and m is move type.

This method is the optimal way to use a mixed predictor set, since no "impossible" N-grams will be estimated. An impossible N-gram has a speaker identity in a move type "slot", for example. However, the software (Taylor *et al.*, 1997a) used to estimate and use the dialogue model restricted the model to one in which all predictors are drawn from the same alphabet (although the predictee alphabet can be different).

Instead of extending the software to implement method 2, an approximation was devised which utilised software tools which were already available – this is method 3. This method will in fact have exactly the same outcome as method 2 since the N-gram models are not smoothed or backed-off.

- *Method 3 : a practical approximation to method 2*

Because it was more practical to estimate models in which all predictors come from the same alphabet, which is different to the predictee alphabet, method 3 was devised. Now, the predictor alphabet is the *union* rather than *product* of the move type and speaker identity alphabets. The predictor alphabet contains the move type set (12 types), the speaker identity set (2 types) and the special symbol representing the start of an utterance: a total of 15 types. The predictee alphabet comprises the move type set plus the special symbol representing the end of an utterance: a total of 13 types. This is suboptimal in that, for each predictor, only a subset of the predictor alphabet is used. This leads to the estimation of some "impossible" N-grams, but this will not matter as, firstly, this will not affect estimation of the useful N-grams, and, secondly, the recogniser will not use these "impossible" N-grams. Now that the alphabet for predictors has been defined, there remains the selection of which items in the dialogue history to use to fill the predictor slots. Some candidate predictors are given in table 6.3, where m_{i-j} is as defined in figure 6.3.

m_{i-1}	s_i
m_{i-2}	s_{i-1}
m_{i-3}	s_{i-2}
m_{i-j}	s_{i-3}

Table 6.3: Candidate predictors

The perplexity of models using various combinations of these predictors were compared. All model probabilities were estimated simply from N-gram frequencies (as in equation 4.1), with no backing-off (see discussion below) or other parameter smoothing. The results are shown in table 6.4 (figures given are on the *test* set).

[1]only the possible ones are counted

Model	Predictors				N-grams[1]	Test set perplexity
I		s_{i-1}	m_{i-1}	s_i	676	5.5
II	m_{i-2}	s_{i-1}	m_{i-1}	s_i	8788	5.8
III		m_{i-2}	m_{i-1}	s_i	4394	6.2
IV		m_{i-j}	s_{i-1}	s_i	676	5.2

Table 6.4: Dialogue model perplexities

- ***Summary of perplexity results***

Table 6.5 summarises the results from table 6.1 and table 6.4 (as elsewhere, perplexity is quoted for the test set). Heterogeneous model IV is the best model. The perplexities of the simple bigram and heterogeneous model III are similar, with the heterogeneous being slightly better due to the addition of current speaker identity as a predictor. The difference is remarkably small, and indicates that either the current speaker identity alone is not a powerful predictor of move type, or model III has too many parameters to be estimated from the training set (the training set contains 10k moves, as shown in table 3.2 on page 39, and model III has 5k parameters).

Model IV is like model I, but replaces the type of the previous move (m_{i-1}) with the type of most recent move by other speaker (m_{i-j}). This reduces the perplexity from 5.5 to 5.2, which indicates that m_{i-j} is a stronger predictor of the current move type than m_{i-1}. This is not surprising, since, in a dialogue, each speaker is generally responding to the other speaker's most recent move. One might guess that j is almost always 1, making $m_{i-1} \equiv m_{i-j}$. This is clearly not the case, as can be seen by examining the training data, where givers have an average 1.3 times as many moves as followers, showing that the giver, at least, takes two successive moves a significant number of times in a dialogue. An example taken from the training data is given below.

participant	move type	words
giver	instruct	"And stop there."
giver	explain	"And right across."
follower	query-w	"Do I go east or west of the well?"
giver	explain	"To the east there should be local residents."
follower	acknowledge	"Right."
follower	reply-y	"Yeah."

- ***The best dialogue model***

The heterogeneous model using current speaker identity (s_i), identity of speaker of preceding move (s_{i-1}) and type of most recent move by other speaker (m_{i-j}) had the lowest perplexity on a held-out portion of the training set (and on the test set, as shown in table 6.5, bottom line, although this figure was not used in the selection of this model), so was chosen as the dialogue model for the full system experiments described in section 7.3.

Model	**Test set perplexity**
simple trigram	6.1
simple 4-gram	6.8
heterogeneous 4-gram (model IV)	5.2

Table 6.5: Dialogue model perplexities (12 move types)

- *The model in detail*

Heterogeneous model IV achieved the lowest perplexity. The predictors seem intuitively reasonable. Clearly current speaker identity has a strong predictive power: giver and follower have quite different distributions of move types. The m_{i-j} predictor is clearly especially appropriate in dialogue situations, where utterances are typically responses to, or follow on from, the other speaker's previous utterance.

We can examine the N-gram frequencies in the chosen model – table 6.6 on page 126 shows a fragment of it. Remember that "!ENTER" is the special symbol for the start of the dialogue, and the speaker (by convention) for "!ENTER" is

always the giver (g). Frequencies of 0 are not shown in the table.

Referring to table 6.6, the most common move types used to start a dialogue are query-yn and ready (lines 7 and 8 of the table), although instruct (line 5) is not uncommon. The most common response by far by the giver to an explain by the follower is acknowledge (line 10). The follower usually answers query-yn with a reply-n or reply-y (lines 28 and 30), with reply-y being twice as likely. This indicates that the giver tends to ask questions which get a positive response.

There are 50 training dialogues, but in the upper section of table 6.6 it can be seen that the number of N-grams with predictors {!ENTER g g} is greater than 50. This is because the giver typically has more than one consecutive move at the start of a dialogue, and my convention is that m_{i-j} is !ENTER if $(i-j) \leq 0$, and that $s_0 = \text{g}$.

- *Backing-off*

On page 58, I discussed the use of backing-off as a technique for estimating longer span N-gram (word) models from sparse data. This technique could also be applied to the estimation of a dialogue model.

The purpose of backing-off is to obtain a better estimate of the probability of low frequency N-grams. A very simple alternative to backing-off is to use a floor probability. This was found to have negligible effect on the perplexity of model IV. Also, the frequencies of N-grams in the chosen dialogue model (see table 6.6), are generally either high, indicating that the model parameters are robustly estimated from the training data, or very low. The negligible effect of the floor probability indicates that those N-grams with zero frequency in the training data really *do not* occur. We can infer that the dialogue model perplexity would not be significantly reduced by a better estimate of the probabilities of those N-grams with zero or very low frequency in the training data. In other words, I would not expect the

predictors m_{i-j}	s_{i-1}	s_i	predictee m_i	frequency	line number
!ENTER	g	g	acknowledge	2	
!ENTER	g	g	align	6	
!ENTER	g	g	check	1	
!ENTER	g	g	explain	7	
!ENTER	g	g	instruct	12	5
!ENTER	g	g	query-w	1	
!ENTER	g	g	query-yn	31	7
!ENTER	g	g	ready	44	8
!ENTER	g	g	reply-y	1	
explain	f	g	acknowledge	220	10
explain	f	g	align	12	
explain	f	g	check	35	
explain	f	g	clarify	4	
explain	f	g	explain	35	
explain	f	g	instruct	43	
explain	f	g	query-w	6	
explain	f	g	query-yn	24	
explain	f	g	ready	26	
explain	f	g	reply-w	2	
explain	f	g	reply-y	3	
query-yn	g	f	acknowledge	45	
query-yn	g	f	check	18	
query-yn	g	f	explain	22	
query-yn	g	f	instruct	1	
query-yn	g	f	query-w	18	
query-yn	g	f	query-yn	16	
query-yn	g	f	ready	3	
query-yn	g	f	reply-n	121	28
query-yn	g	f	reply-w	67	
query-yn	g	f	reply-y	213	30

Table 6.6: A fragment of the chosen dialogue model

perplexity of the chosen model to be significantly improved through backing-off.

Furthermore, backing-off becomes more complicated when the predictors are of mixed types, since the choice of which predictor to drop is not a clear one. This raises another interesting point. For word N-gram models, the decision to drop the leftmost ("oldest") predictor is based on intuition rather than experimental evidence. I suggest that the choice of which predictor to drop when backing-off should be based on training data. For example, the choice should result in the lowest perplexity model on held-out data.

Given that the choice of which predictor to drop can be made experimentally, a longer span dialogue model could have been trained through the use of backing-off techniques. A simple trigram model has around three thousand parameters to be trained from only 10 thousand tokens and has a higher (worse) perplexity than model IV; this indicates that the data is too sparse to train a simple trigram model. A proper investigation of the problems touched on above is outside the scope of this thesis, and since I do not expect backing-off would have a significant effect on model IV, I will restrict the dialogue model to be a non-backed-off N-gram model.

Chapter 7

System performance

7.1 Introduction

The component parts of the experimental system have been developed in the preceding chapters: a speech recogniser, a dialogue model and a set of utterance type-specific language models. In this chapter, I describe the integration of these components, and I will show that the performance of the system is better, in terms of word accuracy, than the baseline speech recogniser. I will also show that the system performs utterance type classification more accurately than any individual component. As already noted, the modular architecture allows testing of individual components, and means that these components can be treated as "black boxes" in the whole system – the method that each component uses to perform its task is not important: for example, the accent detector might be neural net or HMM based. The interfaces between the modules are simple – either the passing of log likelihoods or pitch accent parameters. Note that from now on, utterances are *moves*, and the two terms will be used interchangeably. The utterance type set is the Map Task set of 12 move types from (Kowtko *et al.*, 1993).

- *Organisation of this chapter*

I begin with a formal derivation of an equation whose solution gives the most likely utterance type sequence for a dialogue. Then I describe various experiments using the strategy from page 6 – namely, find the most likely utterance types for a sequence of utterances (i.e. a dialogue), then determine the word strings for each utterance. Conclusions about the results in this chapter can be found in chapter 8. The strategy used was outlined at the end of chapter 2, and the DCIEM corpus used in all experiments was described in section 6.3.

- *Division of labour*

The modular system architecture allows development of each component in parallel. This architecture is indeed essential, since the various components are not all the work of the same person. To repeat from page 9: the accent detection and parameterisation component was the work of Paul Taylor (1992; 1993; 1994); the utterance type intonational HMMs were the work of Helen Wright (Wright & Taylor, 1997); all other components (speech recogniser, language models, dialogue model and system integration) were the work of the author.

- *Division of computation*

It is worth noting that the only part of the system which is seriously computationally expensive is the actual speech recognition, not least because the recogniser has to be run once for each move type on each utterance. The detection of accents and classification of intonational patterns into move types, and the Viterbi search through the dialogue, are trivial by comparison. Typically, for each move in a dialogue, speech recognition took 2 1/2 minutes (in total, for 12 runs per move); accent detection and labelling took about 1 second; intonational tune-to-type recognition took only 12 seconds per dialogue. The Viterbi search took

approximately 13 seconds per dialogue (the average number of moves in each of the five test dialogues is 212). These times are for a Sun Ultra 1.

- ***Implementation***

The software used for the experiments was a mixture of code written specially using the freely available Edinburgh Speech Tools (EST) written by Taylor, King and Black (1997a), the commercial HMM toolkit HTK from Entropic (Young *et al.*, 1996) and the public domain CMU language modelling toolkit from Rosenfeld and Clarkson[1] (1997).

Both the EST and the CMU toolkit were used for estimating language models and their perplexities. The CMU toolkit supports several backing-off methods and was preferred for language model estimation. The EST generates language models in a compact form and allows differing predictor/predictee alphabets. The EST Viterbi search algorithm was modified to use these mixed alphabet language models, and to allow certain predictors to be "given" (such as speaker identity). The EST was therefore used to estimate the dialogue model and perform dialogue decoding to find the most likely move type sequence.

HTK effectively limits N-gram language models to bigrams. This limitation could have been overcome by writing additional tools[2]. However, since the speech recognition component already takes much more computation than the rest of the system[3], and because trigram language models would increase the computational requirement (yet are not guaranteed to give better results, given the small training set) over that for bigram models, this was not thought to be worthwhile.

[1]Many thanks to Philip Clarkson for prompt bug-fixing of the toolkit.
[2]HTK file format documentation permitting.
[3]Computation time is closely related to language model complexity.

7.2 Formal derivation

I will derive an equation for finding the most likely utterance type sequence for a dialogue. Move types, as already described, are used for the utterance type system, but the derivation is actually independent of the particular system chosen – provided that the system assigns exactly one type to each utterance.

7.2.1 Notation

D		the dialogue
U		the sequence of utterances in D
N_U		the number of utterances in D
u_i		the ith utterance of D
U	$\equiv$	$\{u_1, u_2, \ldots u_{N_U}\}$
C		spectral observations, e.g. cepstra, for D
c_i		spectral observations for utterance u_i
C	$=$	$\{c_1, c_2, \ldots c_{N_U}\}$
F		other acoustic observations, such as F_0 and energy; I will call this simply *intonation*.
f_i		intonation of utterance u_i
F	$\equiv$	$\{f_1, f_2, \ldots f_{N_U}\}$
$\mathbf{W}$		the word sequence for D
W_i		the word sequence for utterance u_i
$\mathbf{W}$	$\equiv$	$\{W_1, W_2, \ldots W_{N_U}\}$

$$
\begin{array}{rcl}
N_{W_i} & & \text{the number of words in } W_i \\
w_{ij} & & \text{the } j\text{th word in } W_i \\
W_i & \equiv & \{w_{i1}, w_{i2}, \ldots w_{iN_{W_i}}\}
\end{array}
$$

$$
\begin{array}{rcl}
M & & \text{the sequence of move types for } D \\
m_i & & \text{move type of utterance } u_i \\
M & \equiv & \{m_1, m_2, \ldots m_{N_U}\} \\
S & & \text{the sequence of speaker identities for } D \\
s_i & & \text{speaker identity for utterance } u_i \\
S & \equiv & \{s_1, s_2, \ldots s_{N_U}\}
\end{array}
$$

$$
\begin{array}{rcl}
\mathcal{M} & & \text{the move type set} \\
\mathcal{M} & \equiv & [\mathcal{M}_1, \mathcal{M}_2, \ldots \mathcal{M}_{N_\mathcal{M}}] \\
m_i & \in & \mathcal{M}
\end{array}
$$

7.2.2 Dependence and independence

- Acoustic model
  - only uses spectral observations
  - independent of speaker, utterance move type, intonation
- Intonation model
  - only uses intonation observations
  - independent of speaker, spectral observations

- Language model
  - dependent on move type
  - independent of across–utterance context, speaker identity
- Dialogue model
  - dependent only on speaker identity sequence

7.2.3 Finding the most likely move type sequence

We want to find the most likely move type sequence $M^\star$, given the speaker identity sequence, spectral observations and intonation by solving:

$$\begin{aligned} M^\star &= \underset{M}{\operatorname{argmax}}\, P(M|S,C,F) \\ &= \underset{M}{\operatorname{argmax}}\, P(M)P(S,C,F|M) \end{aligned}$$

because $P(S,C,F)$ is a constant for a given D. The independence assumption that speaker identity has no effect on spectral or intonation observations is clearly false, but we already make this assumption in our *speaker-independent* models for speech and intonation recognition. Assuming that S, C and F are independent:

$$\begin{aligned} &= \underset{M}{\operatorname{argmax}}\, P(M)P(S|M)P(C|M)P(F|M) \\ &= \underset{M}{\operatorname{argmax}}\, P(S)P(M|S)P(C|M)P(F|M) \end{aligned}$$

and since $P(S)$ is a constant for any given D – that is, independent of M:

$$= \underset{M}{\operatorname{argmax}}\, P(M|S)P(C|M)P(F|M) \qquad (7.1)$$

Now I will take each of the terms in equation 7.1 in turn. The first term, $P(M|S)$, of equation 7.1 is given by the dialogue model. We can incorporate the word sequence $\mathbf{W}$ into $P(C|M)$. Letting $\mathbf{W}'$ range over all possible word sequences:

$$\begin{aligned} P(C|M) &= \sum_{\mathbf{W}'} P(C|\mathbf{W}')P(\mathbf{W}'|M) \\ &\approx \max_{\mathbf{W}'} P(C|\mathbf{W}')P(\mathbf{W}'|M) \end{aligned} \tag{7.2}$$

where the replacement of summation by maximisation is a change from total likelihood to maximum likelihood. Intuitively, we can be reasonably certain that

$$\sum_{\mathbf{W}'} P(C|\mathbf{W}')P(\mathbf{W}'|M) \propto \max_{\mathbf{W}'} P(C|\mathbf{W}')P(\mathbf{W}'|M)$$

for a given M. The value of $\mathbf{W}'$ which solves equation 7.2 is simply the result of speech recognition; this value is $\mathbf{W}$.

$$P(C|M) \approx P(C|\mathbf{W})P(\mathbf{W}|M)$$

where

$$P(C|\mathbf{W}) = \prod_{i=1}^{N_U} P(c_i|W_i) \tag{7.3}$$

which is given by the HMMs in the speech recogniser, and

$$P(\mathbf{W}|M) = \prod_{i=1}^{N_U} P(W_i|m_i) \tag{7.4}$$

which is given by the move type-specific language models. The third term of equation 7.2 is simply the intonation model

$$P(F|M) = \prod_{i=1}^{N_U} P(f_i|m_i) \tag{7.5}$$

So equation 7.1 is:

$$M^\star = \underset{M}{\text{argmax}} \left\{ \underbrace{P(M|S)}_{\substack{\text{dialogue} \\ \text{model}}} \cdot \underbrace{P(C|M)}_{\substack{\text{speech} \\ \text{recogniser}}} \cdot \underbrace{P(F|M)}_{\substack{\text{intonation} \\ \text{model}}} \right\} \tag{7.6}$$

which, using equations 7.3, 7.4 and 7.5 becomes

$$= \underset{M}{\text{argmax}} \left\{ \underbrace{P(M|S)}_{\substack{\text{dialogue} \\ \text{model}}} \prod_{i=1}^{N_U} \left(\underbrace{\max_{W_i} P(c_i|W_i)P(W_i|m_i)}_{\substack{\text{speech} \\ \text{recogniser}}} \underbrace{P(f_i|m_i)}_{\substack{\text{intonation} \\ \text{model}}} \right) \right\} \tag{7.7}$$

- ***Practical solution***

In the case of the dialogue model, true probabilities can be estimated because the space of possibilities is finite and discrete (it is the move type set $\mathcal{M}$). This is only true because the dialogue is pre-segmented into utterances, which means that the move type sequence is of a fixed length. If the dialogue were not already segmented, the optimisation expressed by equation 7.7 would additionally involve a search for the optimum segmentation of the dialogue into moves. The second and third terms in equation 7.6 cannot be directly estimated. In practice, we can only estimate *likelihoods*, and these will be represented in the log domain. Also, we wish to account for the differing reliability of the estimates. Therefore, the estimates of the terms in equation 7.7, in the form of log likelihoods will be weighted. We can consider these weights to be measures of the relative contribution, or "importance" of each term, reflecting the reliability of the various estimators of move type. If, for instance, the dialogue model is a particularly good predictor of utterance type,

then it will have a higher weighting.

Equation 7.6 can be rewritten as:

$$M^\star \;=\; \underset{M}{\operatorname{argmax}} \left\{ w_D L^D \;+\; \sum_{i=1}^{N_U} \left(w_S L_i^S \;+\; w_I L_i^I \right) \right\} \tag{7.8}$$

where L^D is the log likelihood from the dialogue model, L_i^S and L_i^I are the log likelihoods for utterance i from the speech recogniser and intonation model respectively along the Viterbi path. w_D, w_S and w_I are the weights for the three terms – of course there are only two degrees of freedom, so one of the three could be omitted (set to unity).

Equation 7.8 can be solved efficiently by the Viterbi (Forney, 1973) algorithm provided the dialogue model is restricted to a finite state model. The combination of probabilities from different information sources via a weighted sum in the log domain is common practice in speech recognition, as well as in integrated approaches to the use of intonation or prosody: Dumouchel and O'Shaughnessy (1993) use microprosodic observation probabilities in this way, as explained on page 98.

The strategy employed is to run the intonation recogniser over each utterance, finding L_i^I for $m_i = \mathcal{M}_1, \mathcal{M}_2, \ldots \mathcal{M}_{N_\mathcal{M}}$. The speech recogniser is also run $N_\mathcal{M}$ times for each utterance, each time with a different one of the move type-specific language models, giving values for L_i^S for each move type. The corresponding pairs of log likelihoods are scaled and summed. The Viterbi algorithm is used to find the most likely utterance type sequence, given the dialogue model, solving equation 7.7.

- *Finding the weights*

The weights in equation 7.8 can be determined experimentally on held out training data. There are two degrees of freedom, and a simple search over a grid of values can be used to find the optimum weight values. The results of this experiment are reported on page 139.

- *An alternative approximation*

Andreas Stolke (personal communication) suggests an alternative to the approximation of equation 7.2: that the sum over all possible word sequences be approximated by summing over an N-best sentence list from the speech recogniser. This is obviously a closer approximation than made here[4], but does require the recogniser to produce N-best lists, which is computationally expensive.

7.3 Integrated system experiments

7.3.1 Move type classification

In chapter 2 I established utterance type classification as one objective of the dialogue speech recogniser, in the belief that by recognising utterance type, word accuracy could be improved. Classification of utterances into types is also an end in itself, for the reasons already mentioned. The experimental strategy was therefore to maximise utterance type classification accuracy, then measure word accuracy.

The system architecture can be seen in figures 7.1 and 7.2. The modules in figure 7.1 provide the input to the system in figure 7.2 on page 139. Interfaces between modules are at the utterance level – that is, the speech recogniser and intonation module process whole utterances at a time. The three information

[4]The two approaches are the same for N-best lists of length 1 !

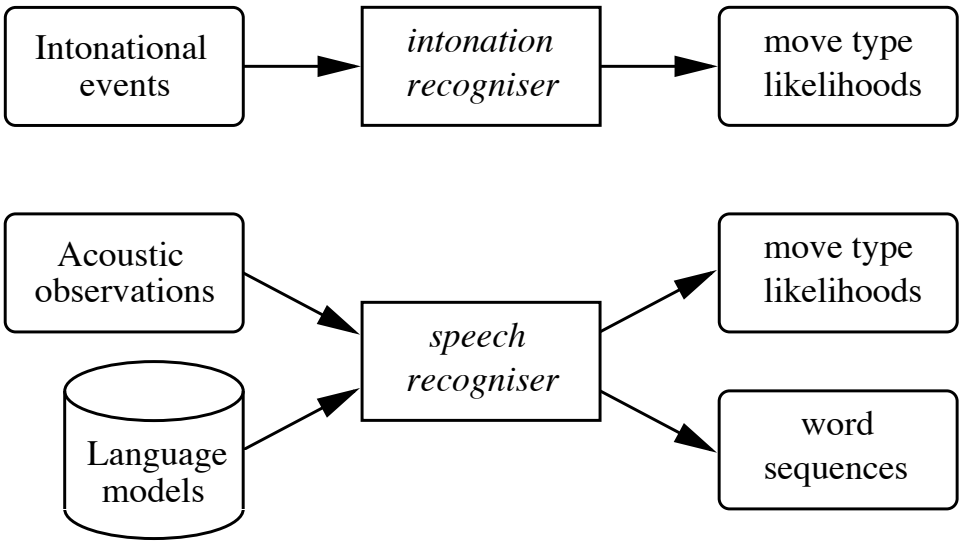


Figure 7.1: System modules

sources used for utterance type classification operate independently – the intonation module does not require the speech recogniser output because it does not use a segmentation of the speech signal.

The core of the system is a Viterbi search for the most likely utterance type sequence for a whole dialogue – a solution to equation 7.6. For a finite state dialogue model, as described on page 124, this means finding the most likely path through the dialogue model, given the observations from the speech recogniser and intonation recogniser. The probability along a path through the network is the product of the transition and state observation probabilities along it. The transition probabilities are given by the dialogue model, and the observation log probability for each state is a weighted sum of speech recognition and intonation recognition log probabilities. Because there are three components making up the total probability (see equation 7.8), there are two weights which can be varied to account for the relative contribution of each information source. Experiments were carried out with various combinations of dialogue model and weights to determine the move type classification power of each component. Since the move type recognition problem is a *classification* task, there are no insertion or deletion

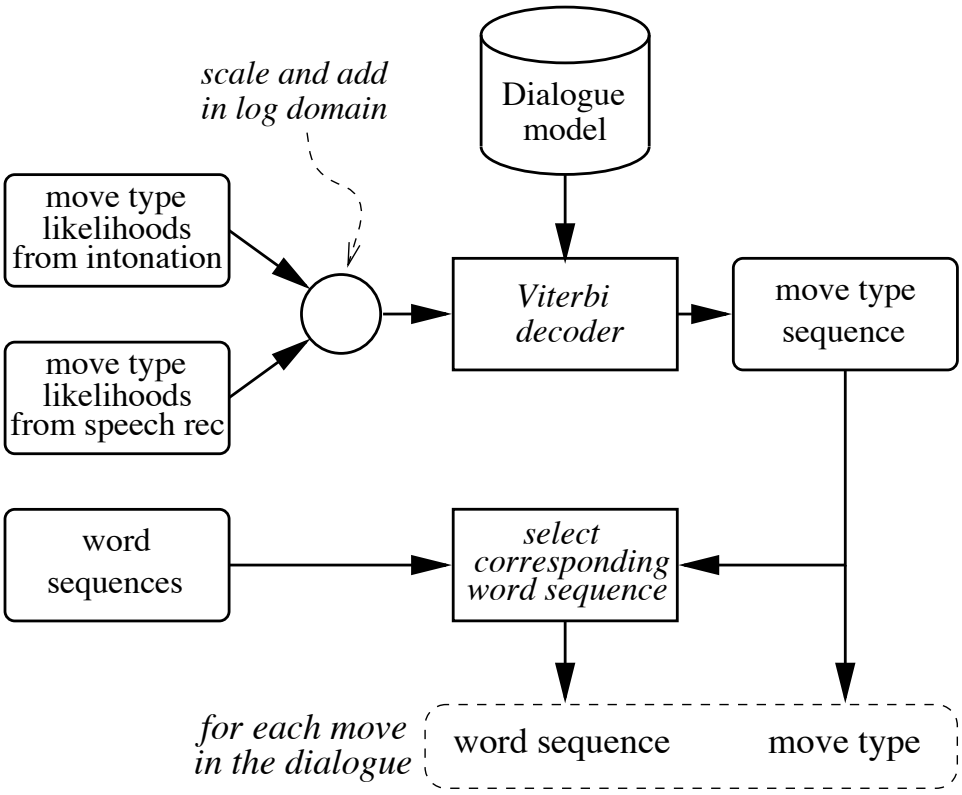


Figure 7.2: System architecture

errors (we assume that a dialogue can be divided into moves perfectly prior to the classification procedure). Therefore, accuracy and percent correct are the same thing.

- *Finding the weights by experiment*

On page 137, I noted that the weights in equation 7.8 on page 136 can be found by experiment. The weights were determined by this method using held-out training data. It is interesting to examine the sensitivity of the move classification accuracy to the weight values. An experiment was carried out to determine this relationship, and the results are shown in figure 7.3. The figure gives move classification results for the test set. The dialogue model weight was fixed at 5 and the acoustic model/language model and intonation weights were varied. Each contour line

in the figure represents a particular move classification accuracy. It can be seen that the weights do not need to be set very precisely to achieve the best move classification result. The weights need only be within ±10% of the optimum to get equally good results.

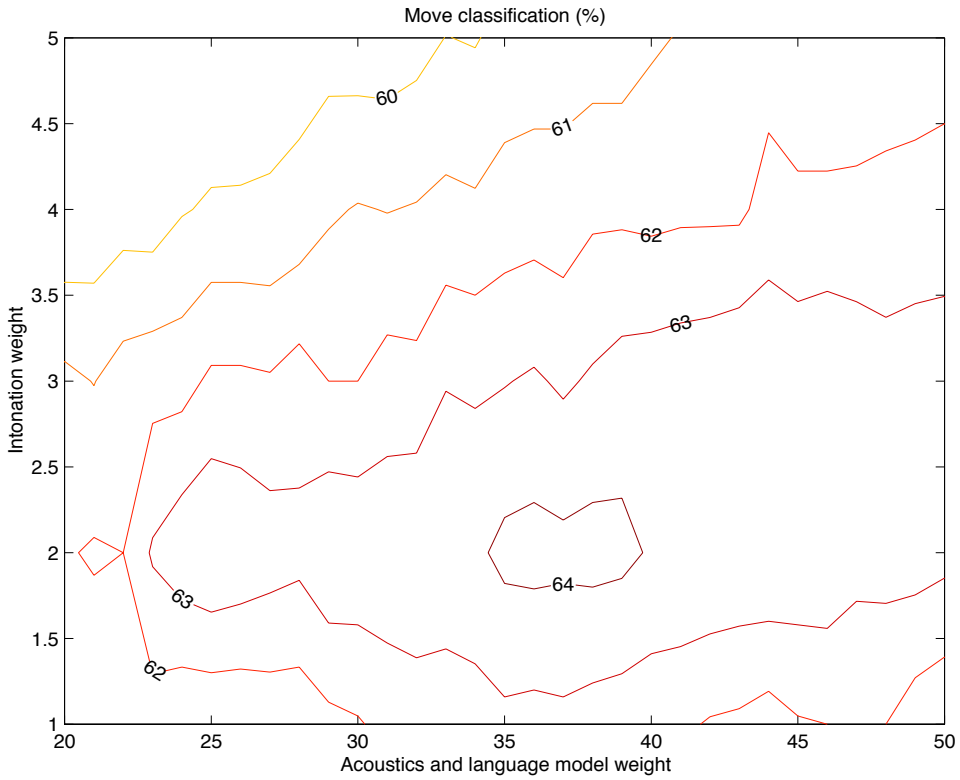


Figure 7.3: Sensitivity of test set move classification results to the choice of weights in equation 7.8

- *Task perplexity*

Although the move type classification problem is a choice of one from twelve, the uneven *a priori* distribution of types means that the task is easier than a choice from 12 equiprobable classes. This distribution can be used to compute the perplexity of the task - a measure of how difficult move type classification is before applying any of the constraints (intonation, speech recognition or dialogue model). The task perplexity is equal to the perplexity of a unigram model, which

condition	word error rate (%)	move type accuracy (%)
Baseline	**24.8**	–
Correct move type every time	23.5	–
Automatic move type classification:		
– acoustics and LM	24.1	40
– acoustics, LM and DM	24.1	57
– intonation only	25.7	42
– intonation and DM	24.7	63
– DM only	–	35
– acoustics,LM,intonation and DM	**23.8**	**64**

Table 7.1: Summarised move type classification and word error rate results. Training data is **set 5.** LM means the language model and DM means the dialogue model.

on the test set is 9.1 (from table 6.1 on page 119).

- ***Using intonation***

It is possible to classify the utterances using only intonation, or intonation plus a dialogue model, without doing speech recognition. This independence of the intonation model from the word or segment sequence is one of its advantages over systems which require a segmentation of the speech signal prior to intonational analysis. Without a dialogue model, intonation classifies 42% of moves correctly, and with the best dialogue model this rises to 63%.

- ***Using speech recognition***

Two varieties of language model (LM) were used: unsmoothed and smoothed. The smoothing was by interpolation with the general language model. As expected, the smoothed models give better word accuracy (see later in this chapter) but do not classify the move types as well as unsmoothed ones. The experiments with both unsmoothed and smoothed language models were performed only on set 4 training/testing data.

For set 5 training data, the "best choice" language model (see page 79) in combination with the best dialogue model (mixed predictor 4-gram model on page 124) correctly classified 57% of moves. Without a dialogue model this figure is only 40%, although word accuracy is good (the error rate is 24.1% – see table 7.1). Since the LM for three of the move types was in fact the general model, it is not surprising that the move type classification power is much lower than the unsmoothed model.

- ***Using only the dialogue model***

It is possible to classify the move types from only the speaker identity sequence using the dialogue model alone. The 4-gram dialogue model correctly classifies 35% of moves. A unigram dialogue model (which does not use speaker identities), gets 24% of move type classifications correct[5].

- ***Intonation and speech recognition together***

By combining the output of intonation and speech recognition components according to equation 7.8, both estimators of move type probability distributions can be used for move type classification. The move type classification results do not improve much on the previous experiment without speech recognition, with

[5]By predicting *acknowledge* all the time, because 24% of moves are *acknowledge*!

64% of moves being correctly classified. However, the word error rate in this case is reduced over the baseline. A summary of results for various combinations of intonation, speech recognition and dialogue model is shown in table 7.1 on page 141.

A confusion matrix for move type classification is shown in table 7.2 for the best combined move recognition and word accuracy condition. Each row details the moves of a particular type. Each column in that row indicates how many moves of that type are classified as each of the 12 possible types. For example, 9 explain moves were classified as check moves. The bold figures along the diagonal are for correctly classified moves. Off-diagonal figures are errors. It can be seen that errors tend to concentrate at certain points – showing that some pairs of move types are more confusible than others. For example, only 2% of align moves are correctly classified since they are mostly misclassified as instruct or ready. The distribution of errors is quite encouraging, since errors appear to be somewhat predictable, rather than random (which would appear as an even distribution of numbers off the diagonal in the confusion matrix).

7.3.2 Speech recognition

When the most likely move type is determined for an utterance, the word string produced by the speech recogniser, using the language model for that type, is taken as the recognised word string for that utterance.

The hypothesis in chapter 2 was that sufficiently accurate move type classification would improve word accuracy results. This is proved correct (refer to table 7.1), although the *method* of move type classification has a significant effect.

If move types are classified using speech recognition alone (which combines move type-specific language model and acoustic models), then only 40% of moves are correctly classified but the word error rate is 24.1% which is better than the

	acknowledge	align	check	clarify	explain	instruct	query-w	query-yn	ready	reply-n	reply-w	reply-y	Correct
acknowledge	**208**	0	1	0	2	2	0	1	28	0	1	16	80%
align	4	**2**	2	0	2	12	0	4	28	1	1	0	3%
check	11	1	**28**	1	1	3	2	13	1	1	3	2	41%
clarify	0	0	0	**7**	0	17	0	0	0	0	3	0	25%
explain	20	1	9	4	**41**	11	0	11	1	6	5	0	37%
instruct	4	1	1	2	6	**172**	0	2	1	0	3	3	88%
query-w	9	0	4	0	1	2	**4**	2	0	0	0	2	16%
query-yn	6	1	13	0	5	5	1	**54**	0	0	1	0	62%
ready	22	0	0	0	1	3	0	1	**46**	1	0	4	58%
reply-n	4	0	0	0	1	0	0	0	0	**23**	1	0	79%
reply-w	3	0	0	2	5	4	1	0	0	0	**6**	2	26%
reply-y	21	1	0	0	3	3	0	1	0	1	2	**76**	70%

Table 7.2: Confusion matrix for move type classification

baseline figure of 24.8%. The low move type classification rate is attributable to the language model smoothing, and so is the low word error rate! With the aid of a dialogue model, the move classification without intonation rises to 57%, but the word error rate remains 24.1%.

Using intonation and a dialogue model, the move recognition rate is 63%, but the word error rate is 24.7%. The best combination of move accuracy and word error rate is obtained when intonation, speech recognition and dialogue model are all used to estimate move type probabilities; move type classification accuracy is then 64% and word error rate 23.8%. This is the lowest word error rate achieved using the novel method introduced in this thesis.

For each utterance, the system generates 12 word sequences. One of these is selected using move type classification. It is interesting to examine one of these lists of word sequence hypotheses – see table 7.3 on page 145.

Correct transcription: *"Go approximately one inch to the left of the telephone booth."*

LM	word sequence
acknowledge	NW go NW approximately NW one inch NW to the left NW of the telephone booth
align	go AB approximately NW one inch NW to the left NW of the telephone booth
check	go NW approximately NW one inch NW to the left NW of the telephone booth
clarify	go AB approximately NW one inch NW to the left NW of the telephone booth
explain	go AB approximately NW one inch NW to the left NW of the telephone booth
instruct	go NW approximately NW one inch NW to the left NW of the telephone booth
query-w	go AB approximately NW one inch NW to the left NW of the telephone booth
query-yn	go AB approximately NW one inch NW to the left NW of the telephone booth
ready	go NW but anyways what inch NW the left NW of the top so booth
reply-n	NW go NW approximately NW when an sheet NW to the left NW on the telephone booth
reply-w	go AB approximately NW one inch NW to the left NW of the telephone booth
reply-y	NW go NW approximate is one inch NW to the left NW of the telephone booth
general	go AB approximately NW one inch NW to the left NW of the telephone booth

Table 7.3: Sentence hypotheses using the 12 smoothed move type-specific language models

- *Significance of results*

The word error rate was reduced from 24.8% to 23.8% by the new method, and a word error rate of 23.5% is possible with 100% move classification accuracy. These error rate reductions were analysed using a paired two-tailed t-test (Iman, 1994). This test requires matching data pairs from two experiments. Here, the pairs being compared comprise the word error rates (for a particular test utterance) for the baseline system and for the new method. The error rates were weighted according to the number of words in the correct transcription of that utterance. The pairs (of utterance word error rates) must be independent of each other. In the baseline and 100% move classification accuracy cases this is a safe assumption, and the error rate reduction from 24.8% to 23.5% is highly significant ($p < 0.0005$).

For the new method, the dialogue model used means that the word error rates for consecutive utterances are no longer independent. As we saw in section 2.2 on page 21, the 12 move types fall into two classes: 6 game initiating types, and 6 other types. It is reasonably safe to assume that within each class, the utterances are independent. This is because, generally, consecutive moves are not of types from the same class. In fact, the improvement in word error rate is entirely due to improvement on game-initiating moves where the baseline word error rate of 26.0% is reduced to 24.7% by the new method, and this reduction is significant ($p < 0.001$). For non-initiating moves, the word error rates increases marginally from a baseline of 19.2% to 19.3% but this difference is statistically insignificant ($p > 0.2$).

condition	word error rate (%)	move type accuracy (%)
Baseline	34.4	–
Correct move type every time	33.5	–
Automatic move type classification:		
– acoustics and LM	33.9	41
– acoustics, LM and DM	34.0	57
– intonation only	34.9	42
– intonation and DM	34.5	64
– DM only	36.1	36
– acoustics,LM,intonation and DM	34.0	64

Table 7.4: Summarised move type classification and word error rate results. Training data is **set 4.**

7.3.3 Effect of training data

Results obtained for the two training sets (set 4 and set 5) can be compared. There is approximately two and a half times more data in set 5 than in set 4 available for training acoustic, language and dialogue models.

- *Language modelling*

Figure 7.4 shows the results from table 3.4 on page 41: the perplexities of general language models trained on various amounts of data. A strong dependency on training set size can be seen, but the perplexity of the language model decreases more slowly as the amount of data increases – it appears to be converging. Therefore, although the data here is very sparse, the difference in perplexity between a back-off bigram model trained on set 4 (27.6) and one trained on set 5 (23.6)

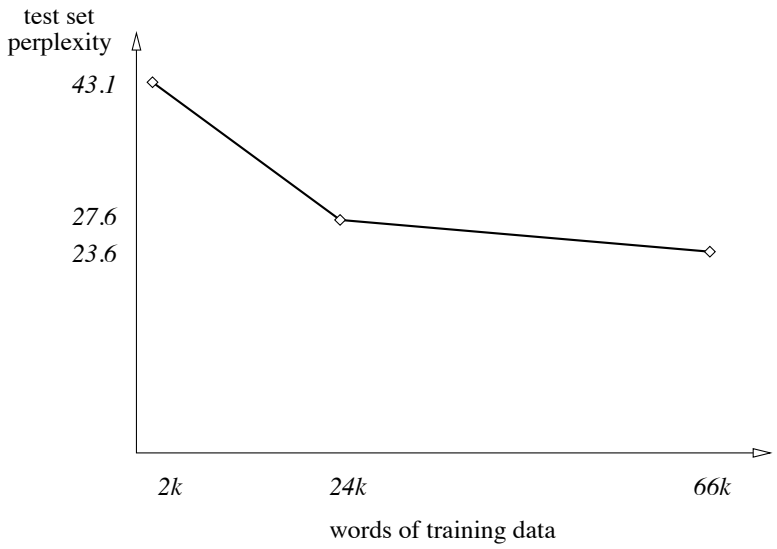


Figure 7.4: Perplexity of language models trained on data sets of various sizes

indicates that perplexity is unlikely to decrease much further even with a large amount more data. However, with more data, more complex models with more parameters can be estimated (a longer span N-gram model for example), which could have lower perplexity.

- *Acoustic modelling*

The acoustic models are tied-state cross-word triphone models. The set of triphones required to cover the vocabulary of the system is independent of the amount of training data. To control the number of parameters in the system, the degree of state tying can be varied. More training data means that more parameters can be estimated, which means that less tying is required – making the HMMs more context-specific which improves acoustic modelling. The tying threshold, which is set during acoustic model training, could be better optimised, but this would be very time consuming.

condition	word error rate (%)	move type accuracy (%)
Baseline	**24.8**	–
Correct move type every time	24.8	–
Automatic move type classification:		
– acoustics and LM	24.5	43
– acoustics, LM and DM	**24.4**	**67**
– intonation only	–	29
– intonation and DM	26.5	52
– DM only	–	45
— acoustics,LM,intonation and DM	**24.4**	**67**

Table 7.5: Summarised move type classification and word error rate results for the alternate set of 13 move types. Training data is **set 5**.

7.3.4 Move type merging and splitting experiments

The alternative set of 13 move types from page 29 was used in exactly the same way as the original 12 types. The same procedures were used for estimating language models, and the experimental system was run in exactly the same way. The results can be seen in table 7.5. The baseline result is, of course, the same as before. Surprisingly, the case where the move type is given (100% correct) produces no improvement in word error rate over the baseline. This indicates that the new set of types either do not form clusters in terms of language model, or there is no longer sufficient training data to estimate language models for this many types. From table 4.4 we can see that, when smoothing the move type-specific language models, the weights for reply-y and reply-n in the original set of 12 move types were close to 1. This means that those language models were well trained and quite different from the general purpose model, implying that a word

error rate reduction would be expected when using these models. In the revised set of 13 move types, reply-y and reply-n were each split into 3 types. Table 4.3 shows that, for reply-n in particular, there is probably not sufficient training data to train language models for the split reply move types.

The automatic detection experiments did produce small word error rate improvements, but not as good as those for the system using the original 12 move types. This error rate reduction is evidently by accident rather than design, since the 100% move classification accuracy experiment gave no error rate reduction at all.

The best move classification accuracy of 67% is marginally better than the result for the original 12 types shown in table 7.1 on page 141, but the corresponding word error rate is worse. The task perplexity[6] using the alternative set of 13 move types is 6.8 which is considerably lower than for the original move type set which has a perplexity of 9.1. This indicates that the task of move type classification for the alternate type set is easier, so the improvement in classification rate from 64% to 67% is poor.

Only one alternate move type classification system was used in the full experimental system. This was the set which gave the most promising results for intonation modelling (Wright, personal communication).

[6]Perplexity of a unigram model on the test set.

Chapter 8

Conclusions

- *Organisation of this chapter*

In this final chapter, I will analyse the results from the previous chapter. I will then describe the limitations of the approach taken, both in terms of the novel method proposed and the experimental work. Then I will describe some possibilities for improvement of the system and its components. Finally, I will suggest how the work described in this thesis might be taken forward.

8.1 Analysis of results

The use of constraints at the utterance level, via utterance type classification, has produced reductions in word error rate along with useful utterance type classification accuracy.

- *Utterance type classification*

The experimental system classified utterances into the 12 Map Task move types with an accuracy of 64%. This figure is high enough to be useful in a dialogue system, where the utterance type class could guide a dialogue manager or syntactic or semantic analysis.

- *Word error rate*

The word error rates achieved by the system are around 24%. This figure is much higher than read-text systems such as those described in section 3.2.1, but compares well with other spontaneous dialogue speech recognisers such as that in (Suhm & Waibel, 1994), Verbmobil – (Plannerer *et al.*, 1994; Reichl & Ruske, 1995), for example – and the Sphinx-II system (Huang *et al.*, 1993) employed in TRAINS.

Assessment of a spoken dialogue system is not easy. Word error rate is not the ideal measure of performance, even for the recogniser component. Typically, the output from the recogniser will be the input to further linguistic and semantic analyses, so raw word error rate is not necessarily an indicator of the recogniser output quality. Boros *et al.* (1996) introduce *concept accuracy* as a better measure of speech recogniser performance, and as a method of simulating the evaluation of the speech recognition component of a spoken language system in isolation. This method requires a parser and semantic processor, so was not suitable for evaluation of the method described in this thesis since these components were not available. One possibly more useful measure of recogniser performance in a spoken language system context, might be key word or content (open class) word accuracy (entities, in the Map Task), although this also has limitations.

- *Limitations of approach*

The utterance type classification approach does have limitations. It can be seen from table 7.1 that even with 100% type classification accuracy, the decrease in word error rate over baseline is from 24.8% to 23.5%, which is a 5.2% reduction in error. This leaves only a small interval for the method described here to operate in, and in fact, the method achieves a 4.0% error rate reduction to 23.8%, which is over three quarters of the possible reduction achievable by 100% move type

classification accuracy.

Since the method achieves an error rate reduction so close to the maximum, the only way to improve the results would be to widen the word error rate interval between baseline and 100% move classification accuracy. This could be done with:

- more training data, especially for language models
- improved utterance type set
- more sophisticated language models and dialogue model

8.2 Analysis of method

The method has shown promising results on a reasonably difficult recognition task. Classification of utterances into types is a general framework for using constraints at the utterance level, and does not depend on any of the particular methods for utterance classification. Intonation, speech recognition and dialogue modelling are just three examples of such methods.

- *Novelty of approach*

Interest in recognition, and understanding, of dialogue speech has risen recently (CLSP, 1997). Use of generalisations above the word level has been on the agenda for speech recognition research for some time, but only recently have serious attempts been made to address the problem (Meteer & Iyer, 1996). This has been due in part to the lack of suitable data; in the early 1990's speech recognition research was driven by the available corpora, the Resource Management Task and Wall Street Journal in particular. With the relatively recent release of the Switchboard and CallHome (Linguistic Data Consortium, 1996a) corpora, interest in spoken dialogue has increased.

With this increase in interest has come the realisation that state-of-the art speech recognisers, such as (Woodland *et al.*, 1995), are limited in their approach when it comes to spontaneous speech. Such systems achieve remarkable performance on clean, read text but do not do nearly as well on real, disfluent speech.

- *Task*

The data used in this work (the DCIEM Map Task corpus) was not recorded specifically for speech recognition research, but the only real alternative available when this work started was the Switchboard corpus, and the reasons for not using that corpus include:

- training a baseline speech recogniser for this task is a major project on its own because
  - there is a very large amount of data to process which makes training acoustic and language models very time consuming
  - the speech is telephone quality (bandlimited)
- the dialogues are only loosely structured
- the dialogue mark-up scheme was only recently defined (following our work here, in fact) and the dialogues annotated
- intonation recognition work is harder on telephone quality speech (Jurafsky *et al.*, 1997)

8.3 Room for improvement

- *Intonation*

The accent detection algorithm used in the system described here is still "work in progress", (Wright & Taylor, 1997). The hidden Markov models of intonational tunes used to classify utterances into move types using accent sequences are also a subject of current work by Wright (Wright & Taylor, 1997). The mapping from intonational events to move types is clearly a very complex one, and in fact may only be possible for some types (Kowtko, 1996).

Since the method of utterance type classification and the classification scheme itself are not specified by the method introduced in this thesis, a variety of algorithms can and have been tried. Wright is currently investigating decision trees and neural nets as alternatives to HMMs.

The intonation component is possibly the most fragile, although, like the speech recogniser, it *is* speaker independent, which is always difficult when dealing with F_0. However, as we have shown, even though the classification power of the intonation component is quite low, it can provide a useful contribution when combined probabilistically with the other components in an utterance classification task.

- *Language modelling*

The move type-specific language models were shown to have lower perplexity than a general-purpose model, especially when smoothed by interpolation with that general model. This smoothing works because the general model was trained on more data (12 times as much on average) than the move type-specific models, and therefore has better estimated parameters.

There is another version of the corpus used here: the original HCRC Map Task, which consists of the same task but with Scottish (Glaswegian) speakers. This corpus could provide significant additional amounts of training data for language modelling, although there are some problems with using it mainly in that the speakers speak a very different kind of English to the Canadian speakers in the DCIEM corpus.

The effects of the mismatch in both vocabulary and grammar can be mitigated – section 4.3.2 described some methods for overcoming such mismatches, but application of them to the DCIEM/HCRC Map Task corpora was beyond the scope of this thesis. One particular problem is that, whilst pronunciation dictionaries for American and Canadian English are readily available, a dictionary would have to be generated specially for the HCRC data.

The language models were conditioned on utterance type, but not on any of the other possible variables: speaker identity, map pair and task condition. Whilst conditioning the models on speaker identity is undesirable, since the system then becomes speaker *dependent* and unable to cope well with new speakers, the map pair and task condition variables are each drawn from a finite set of possibilities and could be used to condition language model probabilities.

Eye contact has a significant effect on the dialogue structure, as described below, and presumably some effect on language. The map used in the dialogue clearly also has an effect, since the entities are different for each map. This situation lends itself to the use of word class models, where the entities across maps are generalised, perhaps into a few broad categories.

- *Speech recognition*

The acoustic models (HMMs) were the same for both baseline and utterance type classification systems – tied-state cross-word triphones. The number of parameters in the models is controlled by the degree of tying. As noted in section 7.3.3, the time for one iteration of training and testing is quite long, optimising the tying thresholds is difficult. It is possible that some improvement could be obtained by fine-tuning the state tying stage of model estimation.

As mentioned below, the HCRC Map Task corpus could provide additional data from the same domain, but in the case of acoustic models, this is of no benefit since the speaker characteristics are so different for the two corpora. Speech recognition work on the HCRC Map Task is currently going on (McKelvie, personal communication).

- *Dialogue modelling*

Just as the language models could benefit from additional training data from the HCRC Map Task corpus, so could the dialogue model. In this case, the mismatch between the two corpora would be much less significant. Indeed, the similarity of dialogue models for the two corpora would provide support for the theory of conversational games, or at least the choice of 12 move types made in (Kowtko *et al.*, 1993).

Another possibility for dialogue model improvement would be to increase the context of the N-gram model, which would mean having to back-off some probabilities. Since the predictors are effectively mixed, this backing-off would be slightly more complex than the simple word N-gram case for language modelling. In backing-off, an N-gram probability is estimated using an (N-1)-gram probability, and defining the (N-1)-gram when the predictors are of mixed types is non-trivial.

As mentioned above, the language models could be conditioned on other variables than utterance type. The presence or absence of eye contact has a significant effect on the dialogue; when eye contact is allowed the task is accomplished more quickly, in fewer moves, and with fewer words. In the HCRC Map Task corpus (Anderson *et al.*, 1991), dialogues conducted without eye conduct contain an average of 13% more words than those with eye contact.

This has implications for human-computer interaction. When the information under discussion is available to both participants, goals can be achieved more quickly – this would be the case in systems like TRAINS where the current "state of play" is represented graphically to the user. For speech-only interfaces (over the telephone, for example), this information sharing may not be possible, and goal oriented dialogues may be less successful. In this situation, one or both participants must take the initiative to set achievable sub-goals in order to achieve the main goal. *Mixed initiative* dialogues are thought by some to be the key to co-operative human-computer interaction.

Another possible extension to the dialogue model would be to use conversational game theory explicitly, in which dialogues are composed of games, and games consist of moves. Such a model would have to account for embedded games as well as terminated moves and games. In other words, it would have to be a *robust* interpretation of the theory of Conversational Games in order to model real world data.

8.4 Further work

- *Multiple language models*

It was found that, unsurprisingly, unsmoothed move type-specific language models had more move type classification power than smoothed ones. However, since smoothed language models had lower perplexity, they were used to achieve lower word error rates. It would be possible to use the unsmoothed models simply as estimators of move type in the same way intonation was used. This does not fit the probabilistic framework of equation 7.7 because the same information is essentially being used twice.

- *The utterance type classification system*

The experiment with the alternate set of 13 move types showed two things. The first was that the new method introduced in this thesis does not depend on the original set of 12 move types – it works with a different set of types too. The second is that, although the alternate type set produced a small improvement over the baseline, it was not as good as that when using the original type set. From this I conclude that the set of 12 move types introduced in (Kowtko *et al.*, 1993) provides a very useful level of description of dialogue motivated utterance types and that these types are distinguished not only by their dialogue rôle, but by their surface form and intonation as well.

References

Allen, J. F., Ferguson, G., Miller, B., & Ringger, E. 1995. Spoken Dialogue and Interactive Planning. *In: Proc. of the ARPA Spoken Language Technology Workshop.*

Allen, J. F., Miller, B. W., Ringger, E. K., & Sikorski, T. 1996 (June). Robust understanding in a dialogue system. *In: Proc. 34th Meeting of the Association for Computational Linguists (ACL '96).*

Alshawi, H. 1992. *The Core Language Engine.* Cambridge, USA: MIT Press.

Anderson, A. H., Bader, M., Bard, E. G., Boyle, E. H., Doherty, G. M., Garrod, S. C., Isard, S. D., Kowtko, J. C., McAllister, J. M., Miller, J., Sotillo, C. F., Thompson, H. S., & Weinert, R. 1991. The HCRC Map Task Corpus. *Language and Speech*, **34**(4), 351–366.

Bahl, L. R., & al. 1988. *Apparatus and method for determining a likely word sequence from labels generated by an acoustic processor.* United States Patent Number 4,748,670.

Bard, E., & Lickley, R. 1997. On not Remembering Disfluencies. *Pages 2855–2858 of: Proc. Eurospeech 97*, vol. 5.

Bard, E. G., Sotillo, C., Anderson, A. H., & Taylor, M. M. 1995. The DCIEM Map Task Corpus: Spontaneous Dialogues under Sleep Deprivation

and Drug Treatment. *In: Proc. of the ESCA-NATO Tutorial and Workshop on Speech under Stress, Lisbon.*

Bartkova, K. 1997 (Sept.). Some experiments about the use of prosodic parameters in a speech recognition system. *Pages 33–36 of: Intonation: Theory, Models and Applications.* ESCA, Athens, Greece.

Beckman, M. E., & Ayers, G. M. 1994 (Feb.). *Guidelines for ToBI Labelling (version 2.0).*

Bennacef, S. K., Néel, F., & Maynard, H. B. 1995 (May). An Oral Dialogue Model based on Speech Acts Categorization. *Pages 237–240 of: Proceedings of the ESCA Workshop on Spoken Dialogue Systems.*

Bird, S., Browning, S., Moore, R., & Russell, M. 1995 (May). Dialogue Move Recognition using Topic Spotting Techniques. *Pages 45–48 of: Proceedings of the ESCA Workshop on Spoken Dialogue Systems.*

Black, A. W., & Campbell, N. 1995 (May). Predicting the intonation of discourse segments from examples in dialogue speech. *Pages 197–200 of: Proceedings of the ESCA Workshop on Spoken Dialogue Systems.*

Boros, M., Eckert, W., Gallwitz, F., Görz, G., Hanrieder, G., & Niemann, H. 1996. Towards understanding spontaneous speech: word accuracy vs. concept accuracy. *In: Proc. ICSLP '96.*

Bruce, G., Granström, B., Gustafson, K., Horne, M., House, D., & Touati, P. 1995 (May). Towards and enhanced prosodic model adapted to dialogue applications. *Pages 201–204 of: Proceedings of the ESCA Workshop on Spoken Dialogue Systems.*

Buder, E. H., & Eriksson, A. 1997. Prosodic cycles and interpersonal synchrony in American English and Swedish. *Pages 235–238 of: Proc. Eurospeech 97*, vol. 1.

Bull, M. 1997. *The timing and coordination of turn-taking.* Ph.D. thesis, University of Edinburgh.

Campbell, N. 1994 (Sept.). Combining the use of duration and F0 in an automatic analysis of dialogue prosody. *Pages 1111–1114 of: Proc. ICSLP-94, Yokohama*, vol. 3.

Campbell, W. N., & Isard, S. D. 1991. Segmental Durations in a Syllable Frame. *Journal of Phonetics*, **19**, 37–47.

Carletta, J., Dahlback, N., Reithinger, N., & Walker, A. 1997a. *Standards for Dialogue Coding in Natural Language Processing.* available from `http://www.georgetown.edu/luperfoy/Discourse-Treebank/dri-home.html` and `http://www.cs.rochester.edu/research/trains/annotation/`.

Carletta, J., Isard, A., Isard, S., Kowtko, J., Doherty-Sneddon, G., & Anderson, A. H. 1995. The coding of dialogue structure in a corpus. *In:* **Andernach, J., van de Burgt, S., & van der Hoeven, G.** (eds), *Proceedings of the Ninth Twente Workshop on Language Technology: Corpus-based Approaches to Dialogue Modelling.* Universiteit Twente, Enschede.

Carletta, J., Isard, A., Isard, S., Kowtko, J., A. Newlands, A., Doherty-Sneddon, G., & Anderson, A. 1997b. The reliability of a dialogue structure coding scheme. *Computational Linguistics*, **23**, 13–31.

Church, K. W., & Gale, W. A. 1991. A comparison of the enhanced Good-Turing and deleted estimation methods for estimating probabilities of English bigrams. *Computer Speech and Language*, **5**, 19–54.

Clarkson, P. R., & Robinson, A. J. 1997. Language Model Adaptation using Mixtures and an Exponentially Decaying Cache. *In: Proc. ICASSP 97.*

CLSP (ed). 1997. *Proc. CLSP/JHU Summer Workshop on Innovative Techniques for Large Vocabulary Conversational Speech Recognition.* Johns Hopkins University, Baltimore. To appear 1998.

Core, M. G., & Schubert, L. K. 1996 (March). *Dialogue parsing in the TRAINS system.* Tech. rept. 612. Computer Science Dept., University of Rochester.

Dogil, G., Kuhn, J., Mayer, J., Möhler, G., & Rapp, S. 1997 (Sept.). Prosody and discourse structure: issues and experiments. *Pages 99–102 of: Intonation: Theory, Models and Applications.* ESCA, Athens, Greece.

Domouchel, P., & O'Shaughnessy, P. 1993 (Sept.). Prosody and continuous speech recognition. *Pages 2195–2198 of: Proc. Eurospeech-93, Berlin*, vol. 3.

Dowding, J., Gowron, J. M., Appelt, D., Bear, J., Cherny, L., Moore, R., & Moran, D. 1993 (June). GEMINI: and natural language system for spoken-language understanding. *In: Proc. 31st annual meeting of the Association for Computational Linguistics.*

Eckert, W., Nöth, E., Niemann, H., & Schukat-Talamazinni, E.-G. 1995 (May). Real Users Behave Weird – Experiences made collecting large Human-Machine-Dialogue corpora. *Pages 193–196 of: Proceedings of the ESCA Workshop on Spoken Dialogue Systems.*

Eckert, W., Gallwitz, F., & Niemann, H. 1996. Combining stochastic and linguistic language models for recognition of spontaneous speech. *Pages 423–426 of: Proc. ICASSP '96*, vol. 1.

Elsner, A. 1997. Focus detection with additional information of phrase boundaries and sentence mode. *In: Proc. Eurospeech 97*, vol. 1.

Essen, U., & Steinbiss, V. 1992 (March). Cooccurence smoothing for stochastic language modeling. *Pages I–161–I–164 of: Proc ICASSP 92*, vol. I.

Ferguson, G. M., Allen, J. F., Miller, B. W., & Ringger, E. K. 1996 (October). *Implementation of the TRAINS-96 System: A Prototype Mixed-Initiative Planning Assistant.* Tech. rept. TRAINS Technical Note 96-5. Computer Science Dept., University of Rochester.

Forney, G. D. 1973. The Viterbi Algorithm. *Proc. IEEE*, March, 268–278.

Garner, P. N., & Hemsworth, A. 1997 (April). A keyword selection strategy for dialogue move recognition and multi-class topic identification. *Page ? of: Proc. ICASSP '97.*

Hess, W., Batliner, A., Kiësling, A., Kompe, R., Nöth, E., Petzold, A., Reyelt, M., & Strom, V. 1996. Prosodic Modules for Speech Recognition and Understanding in VERBMOBIL. *Chap. 23, Part IV, pages 363–383 of:* **Sagisaka, Y., Campbell, N., & Higuchi, N.** (eds), *Computing Prosody, Approaches to a computational analysis and modeling of the prosody of spontaneous speech.* New York: Springer-Verlag. Also available from `http://cl1.ikp.uni-bonn.de/~vst/`.

Hirose, K., & Sakurai, A. 1996. Detection of syntactic boundaries by partial analysis-by-synthesis of fundamental frequency contours. *Pages 809–812 of: Proc. ICASSP '96*, vol. II.

Hirose, K., Sakurai, A., & Konno, H. 1994 (Sept.). Use of prosodic features in the recognition of continuous speech. *Pages 1123–1126 of: Proc. ICSLP-94, Yokohama*, vol. 3.

Hirschberg, J., Nakatani, C., & Grasz, B. 1995 (May). Conveying Discourse Structure through Intonation Variation. *Pages 189–192 of: Proceedings of the ESCA Workshop on Spoken Dialogue Systems.*

Hockey, B. A., Rossen-Knill, D., Spejewski, B., Stone, M., & Isard, S. 1997. Can you predict responses to yes/no questions? Yes, no and stuff. *Pages 2267–2270 of: Proc. Eurospeech 97*, vol. 4.

Huang, X. D., Alleva, F., Hon, H. W., Hwang, M. Y., Lee, K. F., & Rosenfeld, R. 1993. The Sphinx-II speech recognition system: and overview. *Computer Speech and Language*.

Hunt, A. 1993 (Sept.). Utilising prosody to perform syntactic disambiguation. *Pages 1339–1342 of: Proc. Eurospeech-93, Berlin*, vol. 2.

Hunt, A. 1996. Training prosody-syntax recognition models without prosodic labels. *Chap. 20, pages 309–325 of:* **Sagisaka, Y., Campbell, N., & Higuchi, N.** (eds), *Computing Prosody, Approaches to a computational analysis and modeling of the prosody of spontaneous speech.* New York: Springer-Verlag.

Iman, R. L. 1994. *A data-based approach to statistics: concise version.* Duxbury.

Isard, S., King, S., Taylor, P. A., & Kowtko, J. 1995. Prosodic Information in a Speech Recognition System intended for Dialogue. *In: IEEE Workshop in speech recognition.*

Jelinek, F. 1969. Fast sequential decoding algorithm using a stack. *IBM J. Res. Develop.*, **13**(Nov.), 675–685.

Jensen, U., Moore, R. K., Dalsgaard, P., & Lindberg, B. 1993 (Sept.). Modeling of intonation contours at the sentence level using CHMMS and the 1961 O'Connor and Arnold scheme. *Pages 785–788 of: Proc. Eurospeech-93, Berlin*, vol. 2.

Jitsuhiro, T., Yamada, T., & Sagayama, S. 1995 (Sept.). Syllabic duration control for vocabulary-free speech recognition. *Pages 15–18 of: Proc. Eurospeech-95, Madrid*, vol. 1.

Jurafsky, D., Bates, R., Jurafsky, D., Coccaro, N., Martin, R., Meteer, M., Ries, K., Shriberg, E., Stolcke, A., Taylor, P. A., & Ess-Dykema, C. V. 1997. Johns Hopkins LVCSR Workshop-97 SWDB Discourse Language Modelling Project Report. *In: CLSP/JHU Summer Workshop on Innovative Techniques for Large Vocabulary Conversational Speech Recognition.* Johns Hopkins University, Baltimore. `http://www.cstr.ed.ac.uk/publications/pending/Taylor_pending_c.s.ps`.

Katz, S. M. 1987. Estimation of Probabilities from Sparse Data for the Language Model Component of a Speech Recognizer. *IEEE Trans. on Acoustics, Speech, and Signal Processing*, **ASSP-35**(3), 400–401.

King, S., Portele, T., & Höfer, F. 1997. Speech synthesis using non-uniform units in the Verbmobil project. *Pages 569–572 of: Proc. Eurospeech 97*, vol. 2.

Kohonen, T., Torkkola, K., Shozakai, M., Kangas, J., & Ventä, O. 1988. Phonetic Typewriter for Finnish and Japanese. *Pages 607–610 of: Proceedings ICASSP'88.*

Kondo, K. 1995 (Sept.). Connected Japanese digit recognition with pitch accent-dependent models. *Pages 23–26 of: Proc. Eurospeech-95, Madrid*, vol. 1.

Kowtko, J. 1996. *The Function of Intonation in Task-Oriented Dialogue.* Ph.D. thesis, University of Edinburgh.

Kowtko, J., Isard, S., & Doherty, G. 1993. *Conversational Games within dialogue.* Tech. rept. HCRC/RP-31. Human Communication Research Centre, Universities of Edinburgh and Glasgow. `http://www.hcrc.ed.ac.uk/publications/rp-31.ps.gz`.

Ladd, D. R. 1996. *Intonational Phonology.* Cambridge Studies in Linguistics. Cambridge University Press.

Lickley, R., & Bard, E. 1996. On not Recognizing Disfluencies in Dialogue. *In: Proceedings of ICSLP '96.*

Linguistic Data Consortium. 1993-7. *Switchboard speech corpus.* `http://www.ldc.upenn.edu/`.

Linguistic Data Consortium. 1996a. *Call Home speech corpus.* `http://www.ldc.upenn.edu/`.

Linguistic Data Consortium. 1996b. *Resource Management speech corpus.* `http://www.ldc.upenn.edu/`.

Ljolje, A., & Fallside, F. 1987a. Modeling of speech using primarily prosodic patterns. *Computer Speech and Language*, **2**(3/4), 185–204.

Ljolje, A., & Fallside, F. 1987b. Recognition of isolated prosodic patterns using Hidden Markov Models. *Computer Speech and Language*, **2**(1), 27–34.

Medan, Y., Yair, E., & Chazan, D. 1991. Super resolution pitch determination of speech signals. *IEEE Trans. Signal Processing*, **39**, 40–48.

Meteer, M., & Iyer, R. 1996 (May). Modeling Conversational Speech for Speech Recognition. *Pages 33–47 of:* **Brill, E., & Church, K.** (eds), *Proc. Conference on Empirical Methods in Natural Language Processing.*

Munteanu, P., Caillaud, B., Serignat, J.-F., & Caulen-Haumont, G. 1997. Automatic word demarcation based on prosody. *In: Proc. Eurospeech 97*, vol. 3.

Ney, H., Essen, U., & Kneser, R. 1994. On structuring probabilistic dependencies in stochastic language modeling. *Computer Speech and Language*, **8**(1), 1–28.

O'Connor, J. D., & Arnold, G. F. 1961. *Intonation of Colloquial English.* 1 edn. Longman.

O'Connor, J. D., & Arnold, G. F. 1973. *Intonation of Colloquial English.* 2 edn. Longman.

Ostendorf, M., Wightman, C. W., & Veilleux, N. M. 1993. Parse scoring with prosodic information : an analysis/synthesis approach. *Computer Speech and Language*, **7**(3), 193–210.

Paul, D. B. 1992. An efficient $A^\star$ stack decoder algorithm for continuous speech recognition with a stochastic language model. *In: Proceedings of DARPA speech and natural language workshop.*

Pereira, F. C., Singer, Y., & Tishby, N. 1996 (March). *Beyond Word N-Grams.* Tech. rept. AT&T Research / The Hebrew University. Available from `http://xxx.lanl.gov/cmp-lg/`.

Pierrehumbert, J. B. 1980. *The Phonology and Phonetics of English Intonation.* Ph.D. thesis, MIT. Published by Indiana University Linguistics Club.

Pierrehumbert, J. B., & Hirschberg, J. 1990. The meaning of intonational contours in the interpretation of discourse. *In:* **Cohen, P. R., morgan, J., & Pollack, M. E.** (eds), *Intentions in Communication.* MIT press.

Pitrelli, J. F., Beckman, M. E., & Hirschberg, J. 1994. Evaluation of prosodic transcription labeling reliability in the ToBI framework. *Pages 123–126 of: ICSLP '94*, vol. 1.

Plannerer, B., Einsele, T., Beham, M., & Ruske, G. 1994 (September). A Continuous Speech Recognition System Integrating Additional Knowledge Sources in a Data-Driven Beam Search Algorithm. *Pages S01–5.1 – S01–5.4 of: Proc ICSLP '94.*

Power, R. 1979. The organization of purposeful dialogues. *Linguistics*, **17**, 107–152.

Rabiner, L., & Juang, B.-H. 1993. *Fundamentals of speech signal processing.* Prentice Hall.

Rayner, M., & al. 1993 (Sept.). Spoken language translation with mid-90's technology : a case study. *Pages 1299–1302 of: Proc. Eurospeech-93, Berlin*, vol. 2.

Reichl, W., & Ruske, G. 1995 (May). A Hybrid RBF-HMM System for Continuous Speech Recognition. *Pages 3335–3338 of: Proc. ICASSP '95.*

Renals, S., & Morgan, N. 1992 (Dec.). *Connectionist probability estimation in HMM speech recognition.* Tech. rept. TR-92-081. International Computer Science Institute.

Robinson, A. J. 1993 (Sept.). A neural network based, speaker independent, large vocabulary, continuous speech recognition system : The WERNICKE project. *In: Proc. Eurospeech-93, Berlin.*

Rosenfeld, R. 1994. *Adaptive statistical language modeling: A maximum entropy approach.* Ph.D. thesis, Carnegie Mellon University.

Rosenfeld, R., & Clarkson, P. 1997. *CMU-Cambridge Statistical Language Modeling Toolkit v2.* `http://svr-www.eng.cam.ac.uk/~prc14/`.

Ross, K., & Ostendorf, M. 1995. A dynamical system model for recognising intonation patterns. *Pages 993–996 of: EUROSPEECH 95.*

Sag, I., & Liberman, M. Y. 1975. The intonational disambiguation of indirect speech acts. *Pages 487–497 of: Proceedings of the Chicago Linguistics Society*, vol. 11.

Shriberg, E., Bates, R., Taylor, P. A., Stolcke, A., Ries, K., Jurafsky, D., Coccaro, N., Martin, R., Meteer, M., & Ess-Dykema, C. V.

1998. Classification of Dialog Acts in Conversational Speech. *Language and Speech*, **pending.**

Shriberg, E. 1994. *Preliminaries to a theory of speech disfluencies.* Ph.D. thesis, Department of Psychology, University of California, Berkeley, CA.

Silverman, K., Beckman, M., Pitrelli, J., Ostendorf, M., Wightman, C., Price, P., Pierrehumbert, J., & Hirschberg, J. 1992. ToBI: a standard for labelling English prosody. *Pages 867–870 of: Proceedings of ICSLP '92*, vol. 2.

Stolke, A., & Segal, J. 1994. *Precise n-gram Probabilites from Stochastic Context-free Grammars.* Tech. rept. TR-94-007. International Computer Science Institute. Available from `http://xxx.lanl.gov/cmp-lg/`.

Strangert, E. 1997. Relating prosody to syntax: boundary signalling in Swedish. *Pages 239–242 of: Proc. Eurospeech 97*, vol. 1.

Strom, V. 1995 (Sept.). Detection of accents, phrase boundaries, and sentence modality in German with prosodic features. *Pages 2039–2041 of: Proc. Eurospeech-95, Madrid*, vol. 3.

Strom, V., Elsner, A., Hess, W., Kasper, W., Klein, A., Krieger, H. U., Spilker, J., Weber, H., & Görz, G. 1997. On the use of prosody in a speech-to-speech translator. *In: Proc. Eurospeech 97.*

Suhm, B., & Waibel, A. 1994 (Sept.). Towards better language models for spontaneous speech. *In: Proc. ICSLP-94, Yokohama.*

Takagi, K., & Itahashi, S. 1995 (Sept.). Effectiveness of pause detection information in the content word detection of spoken dialogues. *Pages 19–22 of: Proc. Eurospeech-95, Madrid*, vol. 1.

Taylor, P. A. 1992. *A Phonetic Model of English Intonation.* Ph.D. thesis, University of Edinburgh. Published by Indiana University Linguistics Club.

Taylor, P. A. 1993 (Sept.). Automatic recognition of intonation from F0 contours using the rise/fall/connection model. *Pages 789–792 of: Proc. Eurospeech-93, Berlin*, vol. 2.

Taylor, P. A. 1994. The Rise/Fall/Connection model of intonation. *Speech Communication*, **15**.

Taylor, P. A. 1998. Analysis and Synthesis of Intonation using the Tilt Model. *forthcoming.*

Taylor, P. A., Shimodaira, H., Isard, S., King, S., & Kowtko, J. 1996. Using Prosodic Information to Constrain Language Models for Spoken dialogue. *In: Proc. ICSLP '96.*

Taylor, P. A., King, S., & Black, A. 1997a. *Edinburgh Speech Tools.* Available from the Centre for Speech Technology Research `http://www.cstr.ed.ac.uk/`. Email {`pault,simonk,awb`}`@cstr.ed.ac.uk`.

Taylor, P. A., King, S., Isard, S., & Wright, H. 1997b (Sept.). Using Intonation to Constrain Language Models in Speech Recognition. *In: Proc. Eurospeech-97, Rhodes.*

Taylor, P. A., King, S., Isard, S., & Wright, H. 1998,pending. Intonation and dialogue context as constraints for speech recognition. *Language and Speech.*

van Donzel, M. E., & van Beinum, F. J. K. 1997 (Sept.). Pitch accent, boundary tones and information structure in spontaneous discourse Dutch. *Pages 313–316 of: Intonation: Theory, Models and Applications.* ESCA, Athens, Greece.

van Heuven, V. J., Haan, J., & Pacilly, J. J. A. 1997. Automatic recognition of sentence type from prosody in Dutch. *In: Proc. Eurospeech 97.*

Veilleux, N. M., & Ostendorf, M. 1992 (April). Probabilistic parse scoring with prosodic information. *Pages II–51–II–54 of: Proc. ICASSP 92*, vol. 2.

Vereecken, H., Vorstermans, A., Martens, J. P., & Coile, B. V. 1997 (Sept.). Improving the phonetic annotation by means of prosodic phrasing. *Pages 179–182 of: Proc. Eurospeech 97*, vol. 1.

Wahlster, W. 1993. Verbmobil - Translation of Face-To-Face Dialogs. *Pages 29–38 of: Proc. Eurospeech 93.*

Ward, W., & Young, S. 1993. Flexible use of semantic constraints in speech recognition. *Pages 49–59 of: Proc. ICASSP '93.*

Warnke, V., Kompe, R., Niemann, H., & Noth, E. 1997 (Sept.). Integrated dialogue act segmentation and classification using prosodic features and language models. *Pages 207–210 of: Proc. Eurospeech 97*, vol. 1.

Wichmann, A., House, J., & Rietveld, T. 1997 (Sept.). Peak displacement and topic structure. *Pages 329–332 of: Intonation: Theory, Models and Applications.* ESCA, Athens, Greece.

Witten, I. H., & Bell, T. C. 1991. The Zero-Frequency Problem: Estimating the Probabilities of Novel Events in Adaptive Text Compression. *IEEE Transactions on Information Theory*, **37**(4).

Woodland, P. C., & Young, S. J. 1993. The HTK tied-state continuous speech recogniser. *Page 2207 of: Proc. Eurospeech.*

Woodland, P., Leggetter, C., Odell, J., Valtchev, V., & Young, S. 1995 (May). The 1994 HTK Large Vocabulary Speech Recognition System. *In: Proc. ICASSP '95.*

Wright, H., & Taylor, P. A. 1997 (Sept.). Modelling intonational structure using hidden Markov models. *Pages 333–336 of: Intonation: Theory, Models and Applications.* ESCA, Athens, Greece.

Young, S., Jansen, J., Odell, J., Ollason, D., & Woodland, P. 1996. *HTK manual.* Entropic.